

# Dynamic Gossip

Hans van Ditmarsch\*    Jan van Eijck†    Pere Pardo‡  
 Rahim Ramezani§    François Schwarzenrüber¶

February 3, 2016

## Abstract

A gossip protocol is a procedure for spreading secrets among a group of agents, using a connection graph. We consider distributed gossip protocols wherein the agents themselves instead of a global scheduler determine whom to call. In this paper the problem of designing and analyzing gossip protocols is given a dynamic twist by assuming that when a call is established not only secrets are exchanged but also telephone numbers. Both numbers and secrets can be represented by edges in a gossip graph. Thus, each call may change both the number relation and the secret relation of the graph. We define various such distributed dynamic gossip protocols, and we characterize them in terms of the class of graphs where they terminate successfully.

## 1 Introduction

Gossip protocols are procedures for spreading secrets among a group of agents in a network wherein the agents are represented by the nodes, and the connections represent the ability of the agents to contact each other. There is a vast literature on gossiping in networks, and there are relations to the study of the spreading of epidemics [9, 4]. In the original set-up [11, 7] the network is a complete graph, which means that all agents can call each other, and one of the key questions was to find a minimal sequence of calls to achieve a state where all agents know all secrets. On the assumption that during a call between two agents they exchange all their secrets, in a complete graph with  $n > 3$  agents,  $2n - 4$  calls are needed for this. Assuming four agents  $a, b, c, d$ , a call sequence for ensuring that all secrets are shared is  $ab; cd; ac; bd$  (where  $ab$  represents a call from  $a$  to  $b$ , etc.). The secrets can be shared in 4 calls. If a fifth agent  $e$  is also present, precede this sequence with  $ae$ , and close off with  $ae$ : 6 calls. In general, two extra calls are sufficient for each additional

---

\*LORIA, CNRS – University of Lorraine, France

†University of Amsterdam & CWI, Netherlands

‡LORIA, CNRS – University of Lorraine, France

§Sharif University of Technology, Iran

¶ENS Rennes, IRISA, France

agent, and we have that the number of calls for  $n$  agents equals  $2(n - 4) + 4 = 2n - 4$ . It is a bit trickier to show that less than  $2n - 4$  calls is not possible: see the original [11], or [10]. Papers like [8, 6] investigate other than complete graphs, for which the minimum number of calls needed to distribute all secrets may be larger.

In the case above, the gossip procedure is regulated by a central authority, but in distributed computing we look for procedures that do not need outside regulation. Distributed gossip protocols wherein knowledge is also explicitly modelled were proposed in [3, 1, 2]. A distributed protocol for gossip spreading is:

### **Any Call**

While not every agent knows all secrets, select a pair  $xy$  with  $x \neq y$  and let  $x$  call  $y$ .

This does not exclude redundant calls. A distributed protocol that avoids some of that redundancy is

### **Learn New Secrets**

While not every agent knows all secrets, select a pair  $xy$  such that  $x$  does not know  $y$ 's secret and let  $x$  call  $y$ .

Questions about protocol execution length get different answers in a distributed setting. For 4 agents, the already mentioned  $ab; cd; ac; bd$  is a length 4 execution sequence of Learn New Secrets, but the Learn New Secrets protocol also has an execution  $ab; ac; ad; bc; bd; cd$  of length 6. One can easily show that for  $n$  agents, any sequence between the minimum of  $2n - 4$  and the maximum of (all possible pairs)  $n(n - 1)/2$  can be realized [3]. But in a distributed protocol we cannot guarantee any of these in advance. The reason for that, is that any call is selected that satisfies the condition for execution. In a possibly more appealing way to put the matter: if  $c$  is selected to make the next call after the first call  $ab$ , how can agent  $c$  know that the first call was  $ab$ , and that she should therefore select agent  $d$  to call? Or how can she even know at all that there has been a first call? She merely wants to make a call to anyone whose secret she doesn't know. In other words,  $cd$  is not guaranteed to be the next call. But only if  $cd$  is the next call, will the protocol finish in the minimum of 4 calls. It is an open question what the expected execution length is of the Learn New Secret Protocol under conditions of random scheduling. But we consider such interesting questions outside the scope of our contribution.

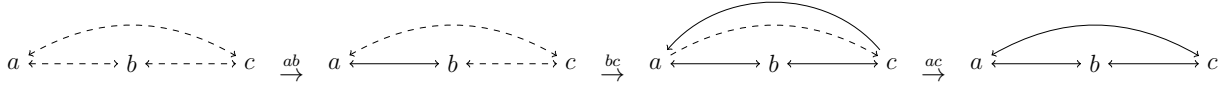
The main point in this paper is that we do not only go distributed, but also go dynamic. In *dynamic gossip* we assume that when a call is established not only secrets are exchanged but also telephone numbers. Calls in the gossip graph are constrained by the current distribution of numbers, so each call may then not only change the distribution of secrets but also change the distribution of numbers. We restrict our scope by assuming that all secrets and all numbers are exchanged, as in standard gossip approaches. Network changing protocols are a standard feature in epidemics [4], but the combination of dynamic gossip and distributed gossip in a non-probabilistic setting is the novel contribution of our paper.

For an example, let us consider three agents only, that agents initially only know their own secret, and initially at least know their own telephone number. A possible protocol is a slightly revised Learn New Secrets:

### Learn New Secrets (with learning numbers)

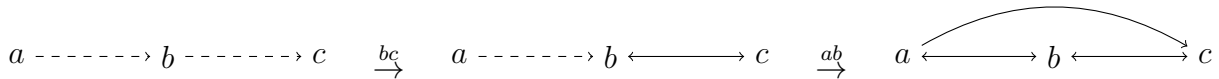
While not every agent knows all secrets, select a pair  $xy$  such that  $x$  knows  $y$ 's number and  $x$  does not know  $y$ 's secret and let  $x$  call  $y$ .

We can model the secrets and numbers of agents as binary relations in a graph. On the assumption that the secret relation is contained in the number relation (you cannot learn someone's secret without knowing her number) we can visualize the number relation as dashed arrows and the secret relation as solid arrows, and further assume reflexivity of relations and that solid is contained in dashed. This is an execution of the Learn New Secrets protocol for three agents in a complete graph:



After the call  $ab$ ,  $a$  and  $b$  know each other's secrets. After the call  $bc$ ,  $b$  and  $c$  know all secrets. Although  $a$  knows  $c$ 's number, it does not know  $c$ 's secret; whereas  $c$  knows  $a$ 's secret: therefore, the arrow for pair  $(a, c)$  is dashed and the arrow for pair  $(c, a)$  is solid. After the final call  $ac$  all agents know all secrets. On complete graphs every execution of the Learn New Secret protocol will terminate successfully.

We now execute the protocol on a graph that is not complete (for the number relation). Without full connectivity, we can execute the same protocol again. On condition that you can only call someone if you have that person's number but do not yet know that person's secret, deadlock is now possible. In the graph below  $a$  knows  $b$ 's number and  $b$  knows  $c$ 's number. (We assume that agents also know their own number.) The call  $bc$  can be made because  $b$  knows  $c$ 's number but doesn't know  $c$ 's secret. After that,  $b$  and  $c$  share their numbers and secrets. Subsequently the call  $ab$  can be made, after which  $a$  and  $b$  know all numbers and all secrets. As they know all secrets, they will make no further calls. Unfortunately, agent  $c$  does not know  $a$ 's secret but also doesn't know  $a$ 's number, so cannot make a call. We are stuck.

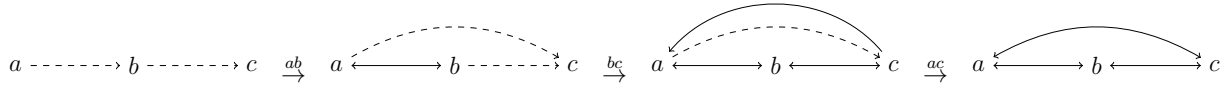


We will show that the Learn New Secrets (LNS) protocol is always successful (which we call *strongly successful*) on graphs when the set of agents (nodes) initially knowing someone else's number is strongly connected (for all such nodes  $x, y$  there is a path from  $x$  to  $y$ ). We call these graphs *sun graphs*. If we restrict the above graph to this set we lose node  $c$  and get

$$a \dashrightarrow b$$

which is not strongly connected, as there is no path from  $b$  to  $a$ . The result is a *characterization*: if graph  $G$  is a sun graph, then LNS is strongly successful on  $G$ , and if LNS is strongly successful on graph  $G$ , then  $G$  is a sun graph.

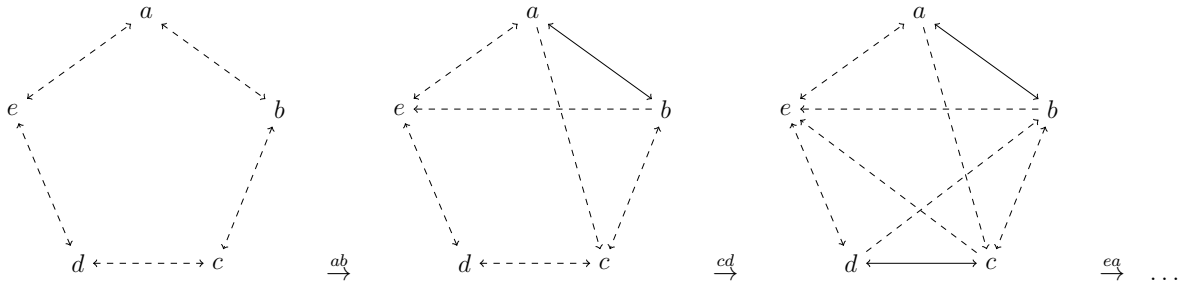
We also define other termination criteria than strongly successful. *Weakly successful* means that there is a (at least one) successful execution of the protocol. For example, even on the graph above there is a successful execution of LNS, namely



We define two large classes of graphs on which LNS is not weakly successful.

We discuss various other protocols exchanging numbers and secrets, such as the Any Call protocol that selects any number to call, which we show to be successful on weakly connected graphs (a graph is weakly connected if there is an undirected path between any two points), and the Call Me Once protocol that only allows calls that you did not make before, that is strongly successful on weakly connected graphs. These are characterization results as well.

**Execution length** Questions on execution length may get different answers in the dynamic gossip context. A well-known and old result is that circle networks of  $n \geq 5$  nodes require  $2n - 3$  calls,  $2n - 4$  is not enough. For example, if we consider the below pentagon, after sequence  $ab; bc; de; cd; ed; ae; bc$  of length 7 all agents know all secrets. In our context, where the network expands during call execution, 6 calls are sufficient  $ab; cd; ea; de; ac; bc$  (of which the first two are demonstrated below): in the dynamic context,  $2n - 4$  is again the minimum number of calls for the circle graph. This proves that the *minimum* length execution of some protocols can be different in our context. As said, this is different from the question what the *expected* execution length of such distributed protocols is, and we defer such matters to future investigation. However, the example may justify the relevance of our characterization efforts.



**Outline** In Section 2 we define gossip graphs, calls, and gossip protocols. The subsequent sections present characterization results for different protocols. Section 3 is on the Any Call protocol, for which success is characterized by weakly connected graphs, and on the Token and Spider protocols that have the same characterization as the Any Call protocol. Section 4 is on the Call Me Once protocol, for which strong success is characterized by weakly connected graphs. These results are all straightforward. Section 5 is devoted to the Learn New Secrets protocol, for which strong success is characterized by sun graphs, and weak success by the complement of graphs that are (what we will later define as) bushes or double bushes. Section 6 summarizes our results, compares the protocols of this paper by their extensions, and suggests further research.

## 2 Gossip Graphs and Gossip Protocols

Given a finite set of agents (or nodes)  $A = \{a, b, \dots\}$  (all lower case letters, possibly quoted or indexed, denote agents), we represent a gossip graph  $G$  with numbers and secrets as a triple  $(A, N, S)$  with  $N, S \subseteq A \times A$ . That is, the agents  $A$  are the vertices and  $N, S$  are binary relations on  $A$ , with  $Nxy$  (for  $(x, y) \in N$ ) expressing that  $x$  knows the (telephone) number of  $y$ , and  $Sxy$  expressing that  $x$  knows the secret of  $y$ .

Let us first introduce standard graph terminology, given a carrier set  $A$  and binary relations like  $N$  and  $S$ . We let  $I_A = \{(x, x)\}_{x \in A}$  be the *identity* relation on  $A$ , and *converse* relation  $N^{-1} = \{(x, y) \mid Nyx\}$ . We write  $N_x$  for  $\{y \in A \mid Nxy\}$ . (We may further write  $\neg Nxy$  for  $(x, y) \notin N$  and anyway write  $xy$  for a pair  $(x, y)$ , such as for the calls defined below.) Relation  $N \circ S = \{(x, y) \mid \text{there is a } z \text{ such that } Nxz \text{ and } Szy\}$  is the *composition* of  $N$  and  $S$ , and using that we define  $N^1 = N$ ,  $N^{i+1} = N^i \circ N$ , and  $N^* = \bigcup_{i \geq 1} N^i$ . Relation  $N$  is *complete* iff  $N = A^2$ , it is *weakly connected* if for all  $x, y \in A$  there is an  $(N \cup N^{-1})$ -path from  $x$  to  $y$ , and it is *strongly connected* if for all  $x, y \in A$  there is an  $N$ -path from  $x$  to  $y$ . For any set  $X$  (like  $A$ , or  $N$ ) we write  $X + x$  for  $X \cup \{x\}$  and  $X - x$  for  $X \setminus \{x\}$ . A pair  $xy \in N$  is a *bridge* iff  $N - xy$  is not weakly connected. Let  $\rightleftharpoons$  be the relation given by:  $x \rightleftharpoons y$  iff there is an  $N$ -path from  $x$  to  $y$  and there is an  $N$ -path from  $y$  to  $x$ . Then  $\rightleftharpoons$  is an equivalence relation, partitioning the domain in strongly connected components. Note that the quotient of a graph  $(A, N)$  with respect to  $\rightleftharpoons$  induces a partial order. A strongly connected component in  $(A, N)$  is *initial* (resp. *terminal*) iff it is a minimal (resp. maximal) element in this partial order.

**Definition 1 (Gossip graph)** A gossip graph is a triple  $G = (A, N, S)$  with  $N \subseteq A^2$  and  $S \subseteq A^2$ . An initial gossip graph is a gossip graph with  $S = I_A \subseteq N$ . Agent  $x$  is an expert if  $S_x = A$ . Agent  $x$  is a spider if  $N_x = A$ . An agent or node is terminal iff  $N_x = \{x\}$ . Gossip graph  $G$  is complete, weakly (strongly) connected if  $N$  is, respectively, complete, weakly (strongly) connected. For  $y \notin G$ ,  $G + y = (A + y, N + yy, S + yy)$ . For  $x, y \in A$ ,  $G + xy = (A, N + xy, S)$ .

In an initial gossip graph each agent only knows its own secret and at least knows its own number. When we employ common graph terminology when referring to a gossip graph,

this applies to the number relation, not to the secret relation, as above.

A *call* from  $x$  to  $y$  is a pair  $(x, y)$  for which we write  $xy$ . The call  $xy$  in  $G$  merges the secrets and the numbers of  $x$  and  $y$ . By  $\overline{xy}$  we mean the call that can be either  $xy$  or  $yx$ .

**Definition 2 (Call)** Let  $G = (A, N, S)$  and  $x, y \in A$ , and  $Nxy$ . The call  $xy$  maps  $G$  to gossip graph  $G^{xy} = (A, N^{xy}, S^{xy})$  defined by

$$N_z^{xy} = \begin{cases} N_x \cup N_y & \text{if } z \in \{x, y\} \\ N_z & \text{otherwise} \end{cases} \quad \text{and} \quad S_z^{xy} = \begin{cases} S_x \cup S_y & \text{if } z \in \{x, y\} \\ S_z & \text{otherwise} \end{cases}$$

The following proposition is immediate:

**Proposition 3** Let  $G = (A, N, S)$  and  $x, y \in A$ , and  $Nxy$ . Then we have:  
 $N^{xy} = N \cup \{(x, y), (y, x)\} \circ N$ , and  $S^{xy} = S \cup \{(x, y), (y, x)\} \circ S$ .

**Definition 4 (Call sequence, Induced gossip graph)** A call sequence  $\sigma$  is a finite or infinite sequence of calls. The empty sequence is  $\epsilon$ . We write  $\sigma; \tau$  for the concatenation of (finite) sequence  $\sigma$  and sequence  $\tau$ . We denote by  $\sigma \sqsubseteq \tau$  that  $\sigma$  is a prefix of  $\tau$  (where  $\sigma \sqsubset \tau$  means proper prefix). We use  $|\sigma|$  for the length of a sequence. Given a finite sequence  $\sigma$  of length  $n$ , and  $i \leq n$ ,  $\sigma[i]$  (for  $i \geq 1$ ) is the  $i$ th call in  $\sigma$ , where for  $\sigma[n]$  we may write  $\text{last}(\sigma)$ , and  $\sigma|i$  is the prefix of  $\sigma$  consisting of the first  $i$  calls. For  $x \in A$ ,  $\sigma_x$  is the subsequence of calls containing  $x$ , defined as:  $\epsilon_x = \epsilon$ ,  $(\sigma; uv)_x = \sigma_x; uv$  if  $x = u$  or  $x = v$ , and  $(\sigma; uv)_x = \sigma_x$  otherwise.

A call  $xy$  is possible given a gossip graph  $G = (A, N, S)$  if  $Nxy$ . Call sequence  $\epsilon$  is possible for any gossip graph. Call sequence  $xy; \sigma$  is possible for  $G$  if call  $xy$  is possible for  $G$  and sequence  $\sigma$  is possible for  $G^{xy}$ . If call sequence  $\sigma$  is possible for gossip graph  $G$ , the induced gossip graph  $G^\sigma$  is defined as:  $G^\epsilon = G$ ,  $G^{xy; \sigma} = (G^{xy})^\sigma$ .

A call sequence  $\sigma$  for  $B$  only involves calls between agents in  $B \subseteq A$ .

The induced gossip graph  $G^\sigma$  is different from the pair  $(G, \sigma)$  consisting of a gossip graph and a possible call sequence for it. Consider the example of the previous section: there are three agents  $a, b, c$  and the number relation  $N$  is the reflexive closure of  $\{(a, b), (b, c)\}$ . Call this gossip graph  $G$ . Consider the possible calling sequence  $ab; bc; ac$  for  $G$ . The first call of this sequence maps  $G$  to  $G^{ab}$ , the second call maps  $G^{ab}$  to  $G^{ab; bc}$ , the third call maps  $G^{ab; bc}$  to  $G^{ab; bc; ac}$ , and the whole sequence maps  $G$  to  $G^{ab; bc; ac}$ . In some protocols it is necessary to refer to call sequences as histories, but the history  $\sigma$  cannot be retrieved from  $G^\sigma$ .

We now come to the definition of *gossip protocol*. A gossip protocol is a program or procedure for selecting calls for execution that satisfy a *protocol condition*.

**Definition 5 (Protocol condition)** Let an initial gossip graph  $G = (A, N, S)$  and a possible call sequence  $\sigma$  be given. A protocol condition is a property  $\pi(x, y)$  that is a boolean combination of constituents  $S^\sigma xy$ ,  $xy \in \sigma_x$ ,  $yx \in \sigma_x$ ,  $\sigma_x = \epsilon$ ,  $\exists z(\text{last}(\sigma_x) = xz)$ , and  $\exists z(\text{last}(\sigma_x) = zx)$ .

This definition of protocol condition is sufficient to justify the protocols treated in this paper. The goal of this paper is to obtain characterization results for those protocols, not to provide a general or formal theory of gossip protocols. Gossip protocol specification languages are investigated in [1, 12].

Many other protocol conditions are conceivable. Informally,  $\pi(x, y)$  may be any first-order definable property with (possibly) free variables  $x$  and  $y$  that satisfies *locality*, i.e., such that agent  $x$  can select agent  $y$  based on the local state of  $x$ , in other words: based on what  $x$  knows. This means that the property may be formulated in terms of:

$N_x^\sigma$ ,  $S_x^\sigma$ ,  $\sigma_x$  (and membership of these sets and that sequence), and for any  $\overline{xy} \in \sigma_x$ ,  $N_y^\tau$  and  $S_y^\tau$  (and membership of those sets), where  $\tau$  is the restriction of  $\sigma$  to the sequence ending in that call  $\overline{xy}$ .

Definition 5 fixes a subset of these options. The protocol conditions used in this work satisfy locality in an obvious way.

**Definition 6 (Gossip protocol)** *A gossip protocol  $P$  with protocol condition  $\pi(x, y)$  is a non-deterministic algorithm of the following shape, operating on any given initial gossip graph  $G = (A, N, S)$ :*

*As long as not all agents are experts, select  $x, y \in A$  such that  $xy$  is possible and satisfies condition  $\pi(x, y)$ , and execute call  $xy$ .*

Alternatively we can describe the gossip protocol as follows. Unlike the above, it does not abnormally terminate —get stuck— in case no possible pair  $xy$  satisfies the protocol condition.

*As long as not all agents are experts and there are  $x, y \in A$  such that  $xy$  is possible and satisfy condition  $\pi(x, y)$ , select  $x, y \in A$  such that  $xy$  is possible and satisfy condition  $\pi(x, y)$ , and execute call  $xy$ .*

The distributed nature of  $P$  protocols with a protocol condition  $\pi(x, y)$  that is local can be seen by reinterpreting the execution procedure as an interpreted system [5] for gossip. For any call sequence  $\sigma$  in  $(A^2)^*$ , let  $\sigma^F$  be the sequence of the first elements of the calls in  $\sigma$ , and let  $D(\sigma) = \{\sigma_a \mid a \in A\}$  be the set of local call histories. Then  $\sigma^F$  and  $D(\sigma)$  together determine  $\sigma$ , and all instantiations of  $\pi(x, y)$  can be checked by agent  $x$  in  $G = (A, N, S)$  on the basis of  $x$ 's numbers and secrets in the initial gossip graph, and the numbers and secrets of the agents involved in the calls in  $\sigma_x$ .

The only aspects of gossip protocols which are not distributed are the selection of an agent  $x$  allowed to make a call (think of this as randomly unblocking the telephone of a single agent for an outgoing call) and the part of the stop condition that involves checking whether all agents are experts, as we cannot even assume that a single agent knows that it is an expert: this requires an agent to know how many agents there are.

The following is yet another consequence of locality, and the distributed nature of our protocols. We assume that all agents follow the same protocol. But this does not entail

that agents know that other agents follow the same protocol. An agent who follows the Learn New Secrets protocol considers it possible that it receives a call from another agent twice during its execution, thus learning that the other agent does not follow that protocol. Of course this will not happen, because the other agent also follows the Learn New Secrets protocol. But the first agent does not know that.

Admittedly, all this is a simplification of distributed computing, for we do not consider the possibility of parallel calls (we leave this for future work), but it permits us to abstract from introducing agents' knowledge in the formal machinery, and it gives us a well-defined class of protocols to consider.

**Definition 7 (Permitted call sequence)** *Let a gossip protocol  $P$  with protocol condition  $\pi(x, y)$  be given.*

- *call  $xy$  is  $P$ -permitted on  $G$  iff  $xy$  is a possible call in  $G = (A, N, S)$ ,  $S \neq A^2$ , and protocol condition  $\pi(x, y)$  holds in  $G$ .*
- *call sequence  $\epsilon$  is  $P$ -permitted on  $G$ .*
- *call sequence  $\sigma; xy$  is  $P$ -permitted on  $G$  iff  $\sigma$  is  $P$ -permitted on  $G$  and  $xy$  is  $P$ -permitted on  $G^\sigma$ .*
- *an infinite call sequence  $\sigma$  is  $P$ -permitted on  $G$  iff for all  $n \in \mathbb{N}$ ,  $\sigma[n+1]$  is  $P$ -permitted on  $G^{\sigma|n}$ .*

If a call sequence is  $P$ -permitted, we also call it a  $P$ -sequence, or a  $P$ -execution.

**Definition 8 (Protocol extension)** *The extension  $P_G$  of protocol  $P$  on  $G$  is the set of  $P$ -permitted sequences on  $G$ . If all call sequences in  $P_G$  are finite (i.e., terminating), protocol  $P$  is terminating on  $G$ . Given protocols  $P$  and  $P'$  and a collection  $\mathcal{G}$  of gossip graphs, we write  $P \subseteq_{\mathcal{G}} P'$  iff  $P_G \subseteq P'_G$  for all  $G \in \mathcal{G}$ , and  $P \subseteq P'$  if  $P \subseteq_{\mathcal{G}} P'$  holds for the collection  $\mathcal{G}$  of initial gossip graphs.*

**Definition 9 (Maximal, stuck, fair)** *Let  $G = (A, N, S)$  and  $P$  be given.*

- *A  $P$ -maximal sequence  $\sigma$  on  $G$  is a sequence  $\sigma$  that is  $P$ -permitted on  $G$  and that is infinite or such that no call is  $P$ -permitted on  $G^\sigma$ .*
- *Given  $B \subseteq A$ , sequence  $\sigma$  is  $P$ -maximal for  $B$  if all calls in  $\sigma$  are between members of  $B$ ,  $\sigma$  is  $P$ -permitted on  $G$ , and  $\sigma$  is infinite or no call between members of  $B$  is  $P$ -permitted on  $G^\sigma$ .*
- *A finite call sequence  $\sigma$  is  $P$ -stuck on  $G$  iff  $S^\sigma \neq A^2$  and  $\sigma$  is  $P$ -maximal on  $G$ .*
- *Call sequence  $\sigma \in P$  is fair on  $G$  iff  $\sigma$  is finite or, whenever  $\sigma$  is infinite, then for all  $xy$ : if for all  $i \in \mathbb{N}$  there is a  $j \geq i$  such that  $xy$  is  $P$ -permitted on  $G^{\sigma|j}$ , then for all  $i \in \mathbb{N}$  there is a  $j \geq i$  such that  $xy = \sigma[j]$ .*



If a call sequence is stuck then no further calls can be made but some agents do not know all secrets. If the context makes clear what the protocol  $P$  is, instead of  $P$ -maximal,  $P$ -stuck,  $P$ -fair, and  $P$ -permitted we write maximal, stuck, fair, and permitted.

**Definition 10 (Success)** *Let a gossip graph  $G$  and a protocol  $P$  be given. A finite call sequence  $\sigma \in P_G$  is successful if in  $G^\sigma$  all agents are experts.*

- *Protocol  $P$  is strongly successful on  $G$  if all maximal  $\sigma \in P_G$  are successful.*
- *Protocol  $P$  is fairly successful on  $G$  if it is strongly successful on the restriction of  $P_G$  to fair call sequences.*
- *Protocol  $P$  is weakly successful on  $G$  if there is a  $\sigma \in P_G$  that is successful.*
- *Protocol  $P$  is unsuccessful on  $G$  if there is no  $\sigma \in P_G$  that is successful.*

*Given a collection  $\mathcal{G}$  of gossip graphs,  $P$  is (strongly, fairly, weakly, un-) successful on  $\mathcal{G}$  iff  $P$  is (strongly, fairly, weakly, un-) successful on every  $G \in \mathcal{G}$ .*

A finite call sequence is fair by definition. If a protocol is fairly successful, then all fair sequences in the extension are finite. Strongly successful implies weakly successful, as the set of maximal call sequences is non-empty (if nothing is permitted, the empty sequence is maximal). If a sequence is successful, it is also maximal. Strongly successful implies fairly successful, as a successful call sequence is finite and thus fair. Fairly successful implies weakly successful, as  $\epsilon$  is fair and each fair call sequence can be extended into a maximal fair call sequence, and therefore the set of maximal fair call sequences is non-empty. So strong implies fair and fair implies weak. Unsuccessful is the same as not weakly successful.

**Definition 11 (Gossip problem)** *Given a collection  $\mathcal{G}$  of gossip graphs and a protocol  $P$ , the gossip problem is: is  $P$  (strongly, fairly, weakly, un-) successful on  $\mathcal{G}$ ?*

It is straightforward to define and implement search algorithms for permitted call sequences and stuck call sequences. A protocol  $P$  is strongly successful on  $G = (A, N, S)$  if either  $S$  is complete, or there is a permitted call  $xy$ , and after any permitted call  $xy$  the  $P$  protocol is strongly successful on  $G^{xy}$ . It follows that  $P$  is strongly successful on  $G$  iff every  $P$ -sequence  $\sigma$  results in a graph  $G^\sigma$  that is complete, or is such that there is a permitted call, and after any permitted call  $xy$ ,  $P$  is strongly successful on  $G^{\sigma;xy}$ . It follows from this definition that a protocol  $P$  is not strongly successful on  $G$  iff there is  $P$ -sequence  $\sigma$  that is infinite or stuck on  $G$ . This gives a straightforward algorithm for recognizing the gossip graphs where  $P$  is strongly successful, provided we know that the protocol terminates:

#### **$P$ gossip graph algorithm**

Search for a stuck call sequence in depth-first fashion, and declare success if no such call sequence can be found [12].

We close this section with some elementary combinatorial results on gossip graphs.

**Proposition 12** *Let  $G = (A, N, S)$  be an initial gossip graph, and let  $\sigma$  be a possible call sequence for  $G$ . Then,  $S^\sigma \subseteq N^\sigma$ .*

**Proof** This is proved by induction on  $\sigma$ . Initially,  $S_x \subseteq N_x$ . Then, it follows from  $S_x^\sigma \subseteq N_x^\sigma$  and  $S_y^\sigma \subseteq N_y^\sigma$  that  $S_x^\sigma \cup S_y^\sigma \subseteq N_x^\sigma \cup N_y^\sigma$ , and therefore  $S_x^{\sigma;xy} = S_y^{\sigma;xy} \subseteq N_x^{\sigma;xy} = N_y^{\sigma;xy}$ .  $\square$

Therefore, it does not matter whether the goal is to learn all secrets, or all secrets and all numbers: if we begin in a situation where we know more numbers than secrets, we cannot learn all secrets without learning all numbers.

**Proposition 13** *Let  $\sigma$  be possible for  $G = (A, N, S)$ . Then,  $G$  is weakly connected iff  $G^\sigma$  is weakly connected.*

**Proof** A possible call is between two agents in the same connected component, and so the callers exchange numbers of agents in the same connected component. Thus, after a possible call, the number of connected components is invariant. Hence  $G$  is weakly connected iff  $G^\sigma$  is weakly connected.  $\square$

**Lemma 14** *If  $G = (A, N, S)$  is an initial gossip graph and  $\sigma$  is a possible call sequence on  $G$ , then  $S^\sigma \circ N \subseteq N^\sigma$ .*

**Proof** Induction on  $\sigma$ . For the base case we have to show that  $S \circ N \subseteq N$ . We have  $S \circ N = I_A \circ N = N \subseteq N$ .

For the induction step, let  $\sigma$  be a possible call sequence, and assume  $S^\sigma \circ N \subseteq N^\sigma$ . Let  $xy$  be a possible call in  $G^\sigma$ .

Let  $(a, b) \in S^{\sigma;xy} \circ N$ . If  $(a, b) \in S^\sigma \circ N$ , then by the induction hypothesis,  $(a, b) \in N^\sigma$ , and hence by  $N^\sigma \subseteq N^{\sigma;xy}$  we get that  $(a, b) \in N^{\sigma;xy}$ , and we are done.

If  $(a, b) \in S^{\sigma;xy} \circ N$  and  $(a, b) \notin S^\sigma \circ N$ , then we may assume (w.l.o.g.) that  $a = x$  and that there is some  $z$  with  $S^{\sigma;xy}xz$ , and  $Nzb$ . From  $S^{\sigma;xy}xz$  it follows that either  $S^\sigma xz$  or  $S^\sigma yz$  (either  $x$  or  $y$  knew the secret of  $z$  before the call  $xy$ ). In the former case, we have  $(x, b) \in S^\sigma \circ N$ , and therefore by the induction hypothesis,  $(x, b) \in N^\sigma$ . In the latter case, we have  $(y, b) \in S^\sigma \circ N$ , and therefore by the induction hypothesis,  $(y, b) \in N^\sigma$ .

From  $(x, b) \in N^\sigma$  or  $(y, b) \in N^\sigma$  it follows by the definition of  $N^{\sigma;xy}$  that  $(x, b) \in N^{\sigma;xy}$ , and we are done.  $\square$

By Lemma 14, the goal of all agents being expert can never be reached in graphs that are not weakly connected.

**Proposition 15** *If  $G$  is not weakly connected, then for any possible call sequence  $\sigma$ ,  $G^\sigma$  is not complete.*

In other words, no such sequence  $\sigma$  terminates successfully on  $G$ . Hence, no protocol is weakly successful on graphs that are not weakly connected. This begs the question what the minimum structural requirements are for protocols to be successful. In the subsequent

sections we will now investigate this systematically for a number of protocols. There exists indeed a protocol CO (call  $xy$  is permitted if neither  $xy$  nor  $yx$  was made before) that is (strongly) successful on weakly connected graphs, so this lower boundary can be met. But there is much beyond that, for example, as mentioned in the introduction, the LNS protocol may get stuck on weakly connected graphs but is successful on sun graphs (‘almost’ strongly connected graphs). And so on. We open the search with the Any Call protocol, whose extension contains that of any other protocol.

### 3 Protocol ANY — Any Call

**Protocol 16 (ANY — Any Call)**  $\pi(x, y) = \top$

*While not every agent knows all secrets, select a pair  $xy$  such that  $x$  knows the number of  $y$ , and let  $x$  call  $y$ .*

The ANY protocol is like the random selection of a call that is usual for gossip protocols in the networks community [4].

For more than two agents, Protocol ANY is not strongly successful on initial gossip graphs. Suppose there are at least three agents  $a, b, c$  and let call  $ab$  be permitted. Then the infinite sequence  $ab; ab; ab; \dots$  is also permitted and agent  $c$  will never learn the secrets of  $a$  and  $b$ .

**Theorem 17** *Protocol ANY is fairly successful on an initial gossip graph  $G$  iff  $G$  is weakly connected.*

**Proof** The  $\Rightarrow$  direction follows from Proposition 15. For the  $\Leftarrow$  direction, we are done if we can show that all fair execution sequences of the protocol are terminating. Let  $G = (A, N, S)$ , and assume that there is a non-terminating execution sequence  $\sigma$ . Let  $x \neq y \in A$ . We show that  $\sigma$  has a prefix  $\tau \sqsubset \sigma$  such that  $S^\tau xy$ .

Since  $G$  is weakly connected, there is a path  $z_0, z_1, \dots, z_n$  with  $y = z_0$  and  $x = z_n$ , and either  $Nz_i z_{i+1}$  or  $Nz_{i+1} z_i$  for all  $i = 0, \dots, n-1$ . As in all such cases a call  $\overline{z_i z_{i+1}}$  is possible (we recall that  $\overline{xy}$  stands for ‘call  $xy$  or call  $yx$ ’), calls  $\overline{z_0 z_1}, \dots, \overline{z_{n-1} z_n}$  will be eventually scheduled in a fair permitted sequence, and in that order, until  $x$  knows the secret of  $y$  after at most  $\overline{z_{n-1} z_n}$  (maybe this moment happened before). There is therefore a  $\tau \sqsubset \sigma$  so that  $S^\tau xy$ . We can now repeat the procedure for any  $z, w \in A$  such that  $S^\tau zw$  does not yet hold. Therefore, there is a  $\tau' \sqsubset \sigma$  such that  $S^{\tau'} xy$  for all  $x, y \in A$ , i.e., all agents are experts. This contradicts the assumption that  $\sigma$  is infinite.  $\square$

**Corollary 18** *Any fair ANY-permitted sequence  $\sigma$  is finite.*

**Corollary 19** *Protocol ANY is weakly successful on an initial gossip graph  $G$  iff  $G$  is weakly connected.*

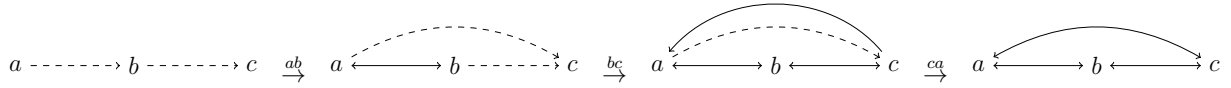
**Proof** If  $G$  is weakly connected, then from Theorem 17 it follows that **ANY** is fairly successful, and therefore weakly successful. If  $G$  is not weakly connected, then it follows from Prop. 15 that **ANY** is not successful, and therefore not weakly successful.  $\square$

We now present two more protocols that are also characterized by weakly connected graphs, and that from that perspective can be seen as variations of the Any Call protocol.

**Protocol 20 (TOK — Token)**  $\pi(x, y) = (\sigma_x = \epsilon \vee \exists z(\text{last}(\sigma_x) = zx))$ .

*While not every agent knows all secrets, agent  $x$  can call agent  $y$  if  $x$  knows  $y$ 's number and ( $x$  has not been in prior calls, or the last call involving  $x$  was **to**  $x$ ).*

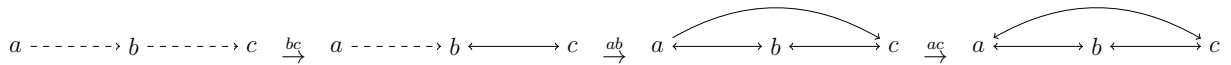
For an example, the sequence  $ab; bc; ca$  is **TOK**-successful in the following graph:



**Protocol 21 (SPI — Spider)**  $\pi(x, y) = (\sigma_x = \epsilon \vee \exists z(\text{last}(\sigma_x) = xz))$ .

*While not every agent knows all secrets, agent  $x$  can call agent  $y$  if  $x$  knows  $y$ 's number and ( $x$  has not been in prior calls, or the previous call involving  $x$  was **from**  $x$ ).*

The sequence  $bc; ab; ac$  is **SPI**-successful in the following graph:



In the Token Protocol, a token is required to make a call; each agent  $x$  is initially endowed with a token, which  $x$  will hand over to the called agent  $y$  after a call  $xy$ . All the tokens received by an agent at a time merge into one token before the next call. In the Spider Protocol, in contrast, it is the called agent  $y$  who hands over the token to the agent  $x$  after a call  $xy$ . Thus, the number of agents who can make calls is non-strictly decreasing in both protocols.

The Spider restriction may seem to make it problematic for information to disseminate through a network: in **SPI**, an agent who has been called will never initiate a call again, and terminal nodes will not call anyone ever. Now consider a potential spider  $x$  weaving its web in a strongly connected part  $X$  of a gossip graph, that also contains another strongly connected component  $Y$  with another potential spider  $y$  weaving its web over there, and a terminal node  $z$  that is connected by a single edge to an element of  $X$  and by a single edge to an element of  $Y$ . It may happen that everybody in  $X$  learns all the secrets of all the agents in  $X$ , and the same among  $Y$ . But as long as  $z$  is not called transfer between  $X$  and  $Y$  is blocked. Eventually  $x$  and  $y$  will call  $z$ , the impasse is broken, and the information will spread.

We continue with the characterization results. For these protocols, fair success is, like for the Any Call protocol, characterized by weakly connected graphs. These results are not hard to obtain. The restriction in the Token protocol is very weak: given a call  $xy$ , call  $yx$  is TOK-permitted, after which  $xy$  is TOK-permitted again; therefore, unless the sequence successfully terminates, any call  $xy$  will occur again in a fair sequence. The restriction in the Spider Protocol is stronger: agents who have been called will not ever make a call again in SPI, so some calls  $xy$  will never be scheduled again.

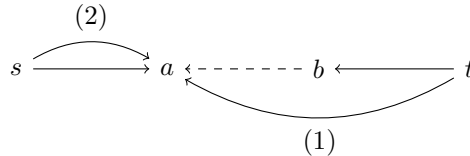
**Theorem 22** *An initial gossip graph  $G = (A, N, S)$  is weakly connected iff Protocol TOK is fairly successful on  $G$ .*

**Proof** For  $(\Leftarrow)$ , use Prop. 15. For  $(\Rightarrow)$ , let  $\sigma$  be a fair TOK-maximal sequence on  $G$ , and let  $x$  be an arbitrary agent. Consider the two cases: (Case 1:  $x$  is not a terminal agent.) Since  $\sigma$  is fair,  $x$  makes a call in  $\sigma$  —initializing a token or being passed one or both. Now every agent with such a token can call agent  $x$ . Since  $\sigma$  is fair,  $x$  will be called arbitrarily often, as long as not all agents are experts. (Case 2:  $x$  is terminal.) Let  $y$  be a predecessor of  $x$ . Then, reasoning as in case 1 but for  $y$ , this agent  $y$  is involved in some token. Because  $Nyx$ , all the agents who received  $y$ 's token can call  $x$ . Since  $\sigma$  is fair and the (increasing) number of agents with  $x$ 's number is finite,  $x$  will eventually be called, and (as in case 1) arbitrarily often so during the protocol execution. In both cases we conclude that all the agents will keep making calls to each of their successors in  $\sigma$ . Therefore any fair TOK-sequence is a fair ANY-sequence. Hence, by Theorem 17,  $\sigma$  is successful on  $G$ .  $\square$

**Theorem 23** *An initial gossip graph  $G = (A, N, S)$  is weakly connected iff Protocol SPI is fairly successful on  $G$ .*

**Proof**  $(\Leftarrow)$  Apply Prop. 15.  $(\Rightarrow)$  It suffices to prove that all infinite SPI-sequences are not fair, so let  $\sigma$  be an infinite SPI-sequence. There is then a prefix  $\tau \sqsubset \sigma$  such that for all  $\tau \sqsubseteq \tau' \sqsubset \sigma$  we have:  $S^{\tau'} = S^\tau \neq A^2$  and  $N^{\tau'} = N^\tau$  and moreover the set of possible callers remains constant after  $\tau$ . Let  $s$  be a possible caller after  $\tau$ . In case  $N_s^\tau = A$ , let  $A = \{s, a_1, a_2, \dots, a_n\}$ . The following pattern of calls will not occur after  $\tau$  in  $\sigma$ , and so  $\sigma$  is not a fair sequence:

$$\tau; \dots; sa_1; \dots; sa_2; \dots; sa_n; \dots; sa_1; \dots; sa_2; \dots; sa_n$$



If  $N_s^\tau \neq A$  then let  $b$  be such that  $\neg N^\tau sb$  and that  $b$  is at minimum  $N$ -distance to the set  $N_s^\tau$ . That is, there is some  $a \in N_s^\tau = S_s^\tau$  such that  $Nab$  or  $Nba$ . If  $Nab$  then the call  $sa$  will not be made after  $\tau$ . Thus,  $\sigma$  is not fair. If  $Nba$  then let  $t$  be a spider in  $G^\tau$  such that  $N^\tau tb$  and  $S^\tau tb$ . By Lemma 14 we have  $N^\tau ta$ . Since  $N^\tau tb$  and  $\neg N^\tau sb$  the pattern

$$\tau; \dots; ta; \dots; sa$$

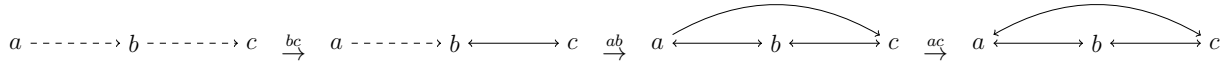
will not occur and so  $\sigma$  is not fair. Now we conclude that all maximal fair SPI-sequences are finite and successful.  $\square$

## 4 Protocol CO — Call Me Once

**Protocol 24 (CO — Call Me Once)**  $\pi(x, y) = xy \notin \sigma_x \wedge yx \notin \sigma_x$

*While not every agent knows all secrets, agent  $x$  may call agent  $y$  if  $x$  knows the number of  $y$  and there was no prior call between  $x$  and  $y$ .*

In other words, agents  $x$  and  $y$  can call each other only *once*. It does not matter who initiated the previous call. Even if  $x$  was the callee and not the caller she is not allowed to become a caller. All CO-permitted call sequences are finite, so fairness is not an issue. But protocol CO is really different from LNS: even if  $x$  has never been in a call with  $y$ ,  $x$  may already have learnt  $y$ 's secret from someone else. This may explain why CO is strongly successful on a larger class of gossip graphs than LNS, and even on the largest conceivable class: weakly connected graphs—we recall that if a graph is not weakly connected, any protocol is unsuccessful. A typical example is the gossip graph of the introductory section, with the LNS-stuck sequence  $bc; ab$ . We can now make another call,  $ac$ , and it terminates successfully.



**Theorem 25** *The CO protocol is strongly successful on an initial gossip graph  $G$  iff  $G$  is weakly connected.*

**Proof** The direction  $(\Rightarrow)$  is immediate (Prop. 15). For the other direction  $(\Leftarrow)$ : Let  $G = (A, N, S)$  be weakly connected and  $\sigma$  be a CO-maximal call sequence. As  $G$  is weakly connected, there is a(n) (undirected) path  $\pi$  in  $G$  between any two agents  $a$  and  $c$ . Assume towards a contradiction that  $c$  does not know the secret of  $a$ . The length of  $\pi$  must therefore be at least 1. Let  $b$  be the first agent on path  $\pi$  who does not know the secret of  $a$  ( $b$  may be  $c$ ). Let  $d$  be the predecessor of  $b$  on the part of  $\pi$  from  $a$  to  $b$  ( $d$  may be  $a$ ). So we have this situation:  $(a, \dots, d, b, \dots, c)$ . By definition of the path,  $Nbd$  or  $Ndb$ . As  $\sigma$  is maximal and not all agents are experts in  $G^\sigma$ ,  $bd \in \sigma$  or  $db \in \sigma$ . Agent  $b$  is the first agent for which  $\neg S^\sigma ba$ , so  $S^\sigma da$ , and therefore  $N^\sigma da$ . Again, as  $\sigma$  is maximal and not all agents are experts in  $G^\sigma$ ,  $da \in \sigma$  or  $ad \in \sigma$ . If  $\overline{ad}$  is before  $\overline{db}$  in  $\sigma$  (we recall  $\overline{xy}$  means ‘ $xy$  or  $yx$ ’) then  $b$  learns the secret of  $a$  from  $d$ , which contradicts  $\neg S^\sigma ba$ . But if  $\overline{db}$  is before  $\overline{ad}$  in  $\sigma$ , then  $a$  knows the number of  $b$  after call  $\overline{ad}$ . We also know that call  $\overline{ab}$  is not in  $\sigma$ , as  $b$  does not know  $a$ 's secret. Therefore, call  $\overline{ab}$  is CO-permitted after  $\sigma$ , which contradicts the maximality of  $\sigma$ .  $\square$

A variant of the protocol CO only requires the same call not to have been made.

**Protocol 26 (wCO — Weak Call Me Once)**  $\pi(x, y) = xy \notin \sigma_x$

*While not every agent knows all secrets, agent  $x$  may call agent  $y$  if  $x$  knows the number of  $y$  and  $x$  did not call  $y$  before.*

**Corollary 27** *The wCO protocol is strongly successful on  $G$  iff  $G$  is weakly connected.*

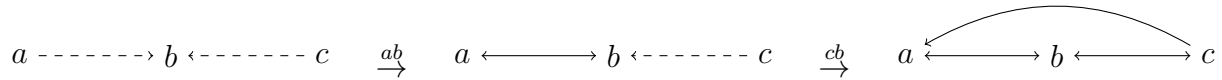
**Proof** The proof is the same as that of Theorem 25, just replace CO by wCO.  $\square$

## 5 Protocol LNS — Learn New Secrets

**Protocol 28 (LNS — Learn New Secrets)**  $\pi(x, y) = \neg S^\sigma xy$ .

*While not every agent knows all secrets, select some  $x$  and  $y$  such that  $x$  knows  $y$ 's number but not  $y$ 's secret and perform the call  $xy$ .*

The LNS protocol has been investigated for complete graphs in [3, 1]. On complete graphs, LNS is strongly successful. The introduction contained an example of a gossip graph on which LNS is weakly successful. On the graph below, LNS is unsuccessful:



The extension of LNS on this graph consists of two executions,  $ab; cd$ , as depicted, and  $cd; ab$ . Both get stuck. This gossip graph is the simplest case of a general problematic class where, whenever an agent  $x$  learns a new number  $z$ , it also learns the secret of  $z$ . To keep LNS going, agents ‘sometimes’ need to learn new numbers without the corresponding secrets, so that they can still make a future call in order to get to know *all* secrets. So it makes sense to ask ourselves which graphs can be completed by LNS.

### 5.1 Where LNS is strongly successful

Employing a number of lemmas and propositions we now characterize the gossip graphs for which LNS is strongly successful. As each agent will make a maximum of  $|A| - 1$  calls, all executions of LNS are finite.

**Proposition 29** *If  $G = (A, N, S)$  is an initial gossip graph and  $\sigma$  is LNS-maximal on  $G$ , then  $S^\sigma = N^\sigma$ .*

**Proof** Let  $G$  be an initial gossip graph, and let there be  $x, y$  with  $N^\sigma xy$  and not  $S^\sigma xy$ . Then the call  $xy$  is permitted in  $G^\sigma$ , which is in contradiction with the maximality of  $\sigma$ . This shows  $N^\sigma \subseteq S^\sigma$ . The property  $S^\sigma \subseteq N^\sigma$  follows from Proposition 12. Together, this gives  $S^\sigma = N^\sigma$ .  $\square$

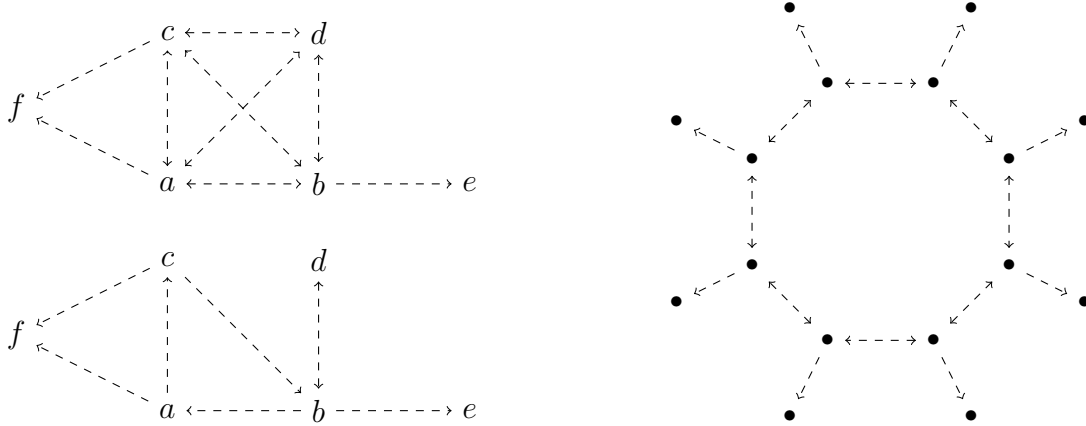


Figure 1: Examples of sun graphs. The restriction to  $\{a, b, c, d\}$  is the strongly connected component for both graphs on the left. Nodes  $e, f$  are terminal. The sun graph on the right demonstrates why we call it a sun.

**Proposition 30** *If  $\sigma$  is an LNS-maximal call sequence on an initial gossip graph  $G$ , then  $S^\sigma \circ N^* = S^\sigma$ .*

**Proof** We have that  $S^\sigma \subseteq S^\sigma \circ N^*$  by definition of  $N^*$ . We now prove that  $S^\sigma \circ N^* \subseteq S^\sigma$ : let  $(x, y) \in S^\sigma \circ N^*$ . Then for some  $k \in \mathbb{N}$ ,  $(x, y) \in S^\sigma \circ N^k$ . We get from Lemma 14 plus Proposition 29 that  $S^\sigma \circ N \subseteq S^\sigma$ . Applying this fact  $k$  times yields  $(x, y) \in S^\sigma$ .  $\square$

The *skin* of a graph  $G = (A, N, S)$  is the set of its terminal nodes  $T_G$ . Let  $s(G)$  be the result of skinning graph  $G$ , i.e. removing all terminal nodes from  $G$ . That is,  $s(G) = (B, N', S')$  where  $B = A \setminus T_G$ , with  $N' = N \cap B^2$  and  $S' = S \cap B^2$ . Skinning a graph is not a closure operation: there are graphs with  $s(s(G)) \neq s(G)$ .

**Definition 31 (Sun graph)** *An initial gossip graph  $G = (A, N, S)$  is a sun graph iff  $N$  is strongly connected on  $s(G)$ .*

We will show that  $G$  is a sun graph if and only if LNS is strongly successful on  $G$ . Figure 1 gives examples of sun graphs.

**Proposition 32** *The LNS protocol is strongly successful on any initial gossip graph  $G$  that is a sun graph.*

**Proof** Let  $G = (A, N, S)$  be a sun graph. Let  $\sigma$  be a LNS-maximal call sequence on  $G$ . Let  $x, y \in A$ . We show that  $S^\sigma xy$ .

If  $x$  is in  $s(G)$ , then  $N^*xy$ . Because of  $Sxx$ , also  $S^\sigma xx$ , and therefore  $(x, y) \in S^\sigma \circ N^*$ . By Proposition 30 it follows that  $S^\sigma xy$ .

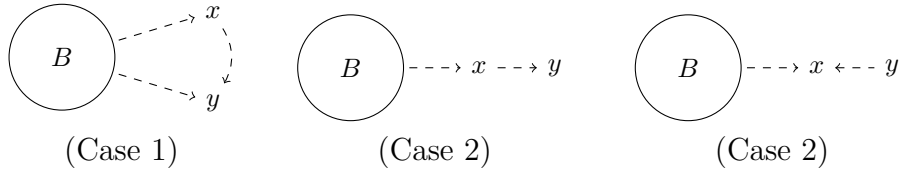
If  $x$  is a terminal node, then by maximality of  $\sigma$ , there is some  $u$  with  $ux \in \sigma$ . This means that  $N^\sigma uz$  for some  $z$  with  $Nzx$  (where possibly  $u = z$ ), because  $u$  must have learnt



$x$ 's number from some such  $z$ . Thus, after the call  $ux$ ,  $x$  has the number of some  $z$  with  $Nzx$ , that is,  $N^\sigma xz$ . By maximality of  $\sigma$  it follows that  $S^\sigma xz$ . Since  $z \in s(G)$  it follows that  $N^*zy$ . From  $S^\sigma xz$  and  $N^*zy$  we get  $(x, y) \in S^\sigma \circ N^*$ . By Proposition 30 we then get  $S^\sigma xy$ , and we are done.  $\square$

**Proposition 33** *Let  $G = (A, N, S)$  be an initial gossip graph. If  $G$  is not a sun then LNS is not strongly successful on  $G$ .*

**Proof** By Prop. 15, we can just assume that  $G$  is weakly connected. Let a subgraph  $H$  of  $G$ , with carrier set  $B$ , be an initial strongly connected component of  $G$ . Let now  $\sigma'$  and  $\sigma''$  be LNS-maximal sequences for  $A \setminus B$  and for  $B$ , respectively, and let  $C$  be the set of agents in  $A \setminus B$  that are successors of agents in  $B$ . We distinguish two cases for which  $G$  is not a sun:



(Case 1) Each agent in  $A \setminus B$  is a successor of an agent in  $B$ . Then,  $C = A \setminus B$ , and there must be agents  $x, y \in C$  such that  $Nxy$  (otherwise  $G$  is a sun). After  $\sigma'$  we have  $S^{\sigma'} xy$ . Let  $Bx$  be a call sequence where everyone in  $B$  calls  $x$ , and let  $\sigma'''$  be a maximal call sequence for  $B \cup (C \setminus \{y\})$ , in graph  $G^{\sigma'; \sigma''; Bx}$ . Observe that  $\sigma'; \sigma''; Bx; \sigma'''$  is LNS-maximal on  $G$ . Clearly agent  $y$  is not expert after this sequence.

(Case 2) There is an agent in  $A \setminus B$  who is not the successor of any node in  $B$ . Then there is an agent  $y \in A \setminus (B \cup C)$ . (The figure above depicts  $\subseteq$ -minimal gossip graphs for Case 2.) Let in this case  $\sigma'''$  be a maximal call sequence for  $B \cup C$ , in the graph  $G^{\sigma'; \sigma''}$ . We now have that  $\sigma'; \sigma''; \sigma'''$  is LNS-maximal on  $G$ . Again, agent  $y$  is not expert.  $\square$

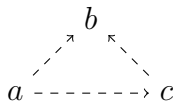
**Theorem 34** *Let  $G = (A, N, S)$  be an initial gossip graph. Then:*

*$G$  is a sun graph iff LNS is strongly successful on  $G$ .*

**Proof** Immediate from Propositions 32 and 33.  $\square$

## 5.2 Where LNS is not weakly successful

Next on our list is weak success. A gossip graph need not be a sun graph in order for LNS to be weakly successful on it, as witnessed by the sequence  $ab; ac; bc$  in the figure below.



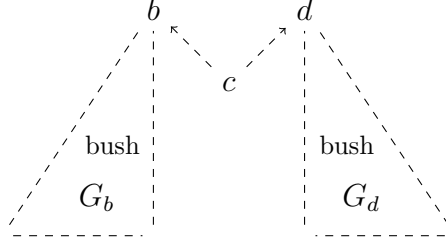
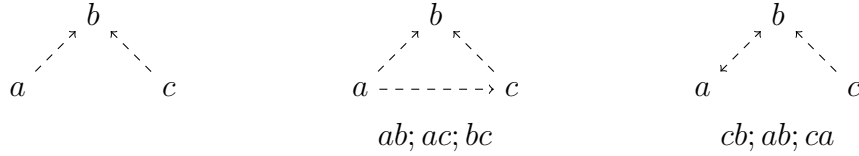


Figure 2: A double bush consists of two bushes linked by a node  $c$  connected to both roots.

It seems hard to find graphs on which **LNS** is unsuccessful, as numerous examples in this section will demonstrate. And once you have found one, add one edge, or one node and one edge, and a successful execution may be possible again. For example, adding a (any) edge to the unsuccessful gossip graph on the left makes it weakly successful, as evidenced by the successful sequence below it:



We define two classes of graphs that are **LNS**-unsuccessful, that of the *bush* and that of the *double bush*. (The unsuccessful one above is a bush.) Our characterization result is then that those are all, so that **LNS** is weakly successful on their complement.

An initial gossip graph  $G = (A, N, S)$  is a *tree* iff  $(A, (N \setminus I_A)^{-1})$  is a directed rooted tree, i.e., from the perspective of the relation  $N \setminus I_A$ , there is unique node  $r$  called *root* such that for any node  $x$  there is a unique directed path from  $x$  to  $r$ . Any node not accessible from other nodes is a *leaf*, and a path from a leaf to the root is a *branch*.

The *in-degree* of a node  $a$  in a gossip graph is the number of incoming  $N$ -edges (excluding loops), i.e.  $\deg_{\text{in}}(a) = |\{b \neq a \mid Nba\}|$ , and the *out-degree* of  $a$  is the number of outgoing  $N$ -edges, i.e.  $\deg_{\text{out}}(a) = |\{b \neq a \mid Nab\}|$ . The *degree* of a node  $x$  is the sum of the in-degree and the out-degree:  $\deg_{\text{io}}(x) = \deg_{\text{in}}(x) + \deg_{\text{out}}(x)$ .

A tree is a *bush* if the in-degree of the root is at least 2. The idea is, that a tree that is not a bush has a trunk. It has in-degree 1.

**Definition 35 (Bush, double bush)** A gossip graph  $G = (A, N, S)$  is a *bush* iff the graph  $(A, N)$  is a bush. A *double bush* consists of two bushes  $G_b + G_d$  that intersect in a leaf  $c$  connected to both roots  $b$  and  $d$ . Given that,  $G$  is a double bush if  $A = A_b \cup A_d$  with  $A_b \cap A_d = \{c\}$  such that the restrictions  $G_b = G|_{A_b}$  and  $G_d = G|_{A_d}$  are bushes and  $\{(c, b), (c, d), (c, c)\} \subseteq N$ . (See Figure 2.)

**Lemma 36** *Let  $G = (A, N, S)$  be an initial gossip graph. If  $G$  is a bush with root  $r$ , and  $\sigma$  is an LNS-sequence in  $G$ , then for any  $x \in A$ :*

1.  $|N_x^\sigma \setminus S_x^\sigma| = \begin{cases} 1 & \text{if } \neg S^\sigma xr \\ 0 & \text{otherwise} \end{cases}$
2. if  $N_x^\sigma \setminus S_x^\sigma = \{z\}$  then  $(N^*xz \text{ and for all } u (N^*zu \text{ implies } \neg S^\sigma xu))$
3. if  $N^*xw, Nwr$  and  $|N_x^\sigma \setminus S_x^\sigma| = 1$ , then for all  $t \in N_x^\sigma$  with  $t \neq r$ , we have  $N^*tw$ .

**Proof** We prove claims 1–3 simultaneously by induction on the length of  $\sigma$ . If  $\sigma = \epsilon$  then the three claims 1–3 hold by Definition 35. For the inductive case of the proof, let us consider  $\sigma; ab$  such that the inductive hypothesis holds for  $\sigma$ .

**Proof of Claim 1**

Claim 1 is obviously true for  $\sigma; ab$  for agents different from  $a$  and  $b$ . Then, for agent  $a$ , we have

$$\begin{aligned}
& |N_a^{\sigma;ab} \setminus S_a^{\sigma;ab}| \\
&= |(N_a^\sigma \cup N_b^\sigma) \setminus (S_a^\sigma \cup S_b^\sigma)| && \text{(by Def. 2)} \\
&= \begin{cases} |(S_a^\sigma \cup \{b\} \cup S_b^\sigma \cup \{z\}) \setminus (S_a^\sigma \cup S_b^\sigma)| & \text{if } N_b^\sigma \setminus S_b^\sigma = \{z\} \\ |(S_a^\sigma \cup \{b\} \cup S_b^\sigma) \setminus (S_a^\sigma \cup S_b^\sigma)| & \text{if } N_b^\sigma \setminus S_b^\sigma = \emptyset \end{cases} && \text{(by Claim 1 for } \sigma) \\
&= \begin{cases} |\{z\}| & \text{if } N_b^\sigma \setminus S_b^\sigma = \{z\} \\ |\emptyset| & \text{if } N_b^\sigma \setminus S_b^\sigma = \emptyset \end{cases} && \text{(by } (\star) \text{ below)} \\
&= \begin{cases} |N_b^\sigma \setminus S_b^\sigma| & \text{if } \neg S^\sigma br \\ 0 & \text{otherwise} \end{cases} && \text{(by Claim 1 for } \sigma) \\
&= \begin{cases} 1 & \text{if } \neg S^{\sigma;ab} ar \\ 0 & \text{otherwise} \end{cases} && \text{(by } (\star\star) \text{ below)}
\end{aligned}$$

( $\star$ ) From  $N_b^\sigma \setminus S_b^\sigma = \{z\}$  it follows that  $N^*bz$  using Claim 2 for  $\sigma$ . From  $N_a^\sigma \setminus S_a^\sigma = \{b\}$  it follows that  $N^*ab$  and for all  $u$  ( $N^*bu \Rightarrow \neg S^\sigma au$ ) (Claim 2). Applying that to  $N^*bz$  we get  $\neg S^\sigma az$ .

( $\star\star$ ) This is because  $S^\sigma br$  iff  $S^{\sigma;ab} ar$ . From left to right is obvious. For the other direction: since the call  $ab$  was made,  $|N_a^\sigma \setminus S_a^\sigma| = 1$ , thus by Claim 1 we have  $\neg S^\sigma ar$ . So the only possible way for  $a$  to learn the secret of  $r$  during the call  $ab$  is that  $S^\sigma br$ . An analogous argument shows Claim 1 for agent  $b$ .

**Proof of Claim 2**

Assume  $N_x^{\sigma;ab} \setminus S_x^{\sigma;ab} = \{z\}$ , and consider the following 3 cases. (Case  $x \notin \{a, b\}$ ) We are done, since  $N_x^{\sigma;ab} = N_x^\sigma$  and  $S_x^{\sigma;ab} = S_x^\sigma$ . (Case  $x = a$ ) We first get  $N_a^\sigma \setminus S_a^\sigma = \{b\}$ , since

the call  $ab$  was permitted after  $\sigma$  and by Claim 1. From that, Def. 2 and again Claim 1, we get  $N_b^\sigma \setminus S_b^\sigma = \{z\}$ . From  $N_a^\sigma \setminus S_a^\sigma = \{b\}$  it follows by induction that:  $N^*ab$  and for all  $u$  with  $N^*bu$ , we have  $\neg S^\sigma au$ , and from  $N_b^\sigma \setminus S_b^\sigma = \{z\}$  it follows by induction that:  $N^*bz$  and for all  $u$  with  $N^*zu$ , we have  $\neg S^\sigma bu$ . From these two we obtain with Def. 2:  $N^*az$  and for all  $u$  with  $N^*zu$ ,  $\neg S^{\sigma;ab} au$ . (Case  $x = b$ ) The argument is analogous.

**Proof of Claim 3**

Suppose  $N^*xw, Nwr$  and  $|N_x^\sigma \setminus S_x^\sigma| = 1$ . Consider again the three cases: (Case  $x \notin \{a, b\}$ ) Obvious. (Case  $x = a$ ) Then,  $b \in N_a^\sigma$  so by the induction hypothesis (Claim 2) we have  $N^*bw$ . Then, since  $N_a^{\sigma;ab} = N_a^\sigma \cup N_b^\sigma$ , in the case  $N_b^\sigma \setminus S_b^\sigma = \emptyset$  we are done, so assume that  $N_b^\sigma \setminus S_b^\sigma \neq \emptyset$ . Then, for all  $t \in N_b^\sigma$  we have  $N^*tw$ . (Case  $x = b$ ) In this case, we have that  $\{b\} = N_a^\sigma \setminus S_a^\sigma$ , so by Claim 2,  $N^*ab$ . The latter, combined with the assumption  $N^*bw$ , gives  $N^*aw$ . Now, by the induction hypothesis, for all  $t \in N_a^\sigma$  we have that  $N^*tw$ . In case  $N_b^\sigma \setminus S_b^\sigma = \emptyset$  we are done. If  $N_b^\sigma \setminus S_b^\sigma \neq \emptyset$  then by the induction hypothesis we also have that for all  $t \in N_b^\sigma$ ,  $N^*tw$ .  $\square$

**Proposition 37** *Let  $G = (A, N, S)$  be a weakly connected initial gossip graph. If  $N$  is a bush, LNS is not weakly successful on  $G$ .*

**Proof** Let  $\sigma$  be LNS-maximal on  $G$ , and let  $yr$  and (subsequently)  $ur$  be the first two calls in  $\sigma$  to the root  $r$  such that  $y$  and  $u$  are in subtrees generated by *different* branches to the root.

To see that  $\sigma$  is not successful, it suffices to prove that for any prefix  $\tau \sqsubseteq \sigma$  it holds that  $\neg S^\tau yu$ . The proof is by induction on  $\tau$ .

The base case is where the last call in  $\tau$  is  $yr$ . Let the sequence without  $yr$  be called  $\tau^-$ . By Lemma 36.1,  $|N_y^{\tau^-} \setminus S_y^{\tau^-}| = 1$  and by Lemma 36.3,  $\neg N^{\tau^-} yu$  and therefore also  $\neg S^{\tau^-} yu$  (Prop. 12). Now,  $S_y^\tau = S_y^{\tau^-} \cup S_r^{\tau^-} = S_y^{\tau^-} \cup \{r\}$  and so  $\neg S^\tau yu$ .

The inductive case is where the last call in  $\tau$  is a call  $ab \neq yr$  (that is,  $ab$  comes sometime after  $yr$ ), let  $\tau^-$  in this case be the sequence without  $ab$ . We distinguish the case  $b \neq y$  from the case  $b = y$ . (Case  $b \neq y$ ) By Lemma 36.1, after the call  $yr$ ,  $y$  will not make further calls. Therefore,  $a \neq y$ . Therefore  $y$  does not learn  $u$ 's secret in the call  $ab$ . Our inductive hypothesis is that  $y$  also did not learn  $u$ 's secret before the call  $ab$ , so therefore we are done. (Case  $b = y$ .) The last call in  $\tau$  is therefore  $ay$ . Applying Lemma 36.1 we get  $N_a^{\tau^-} \setminus S_a^{\tau^-} = \{y\}$ . Therefore, by Lemma 36.2, there must be a path in the tree connecting  $a$  and  $y$ , i.e.  $N^*ay$ . From that and Lemma 36.3 follows  $\neg N^{\tau^-} au$ , and therefore also (Prop. 12)  $\neg S^{\tau^-} au$ . The induction hypothesis is that  $\neg S^{\tau^-} yu$ . As neither  $a$  nor  $y$  know the secret of  $u$ , they will not learn it either in the call  $ay$ :  $\neg S^{\tau^-;ay} yu$ .  $\square$

**Proposition 38** *Let an initial gossip graph  $G = (A, N, S)$  be a double bush. Then the protocol LNS is not weakly successful on  $G$ .*

**Proof** Let  $b$  and  $d$  be the two roots of  $G$  and let  $\sigma$  be a maximal LNS-sequence. Without loss of generality, assume that the first call to  $b$  takes place before the first call to  $d$ . Let

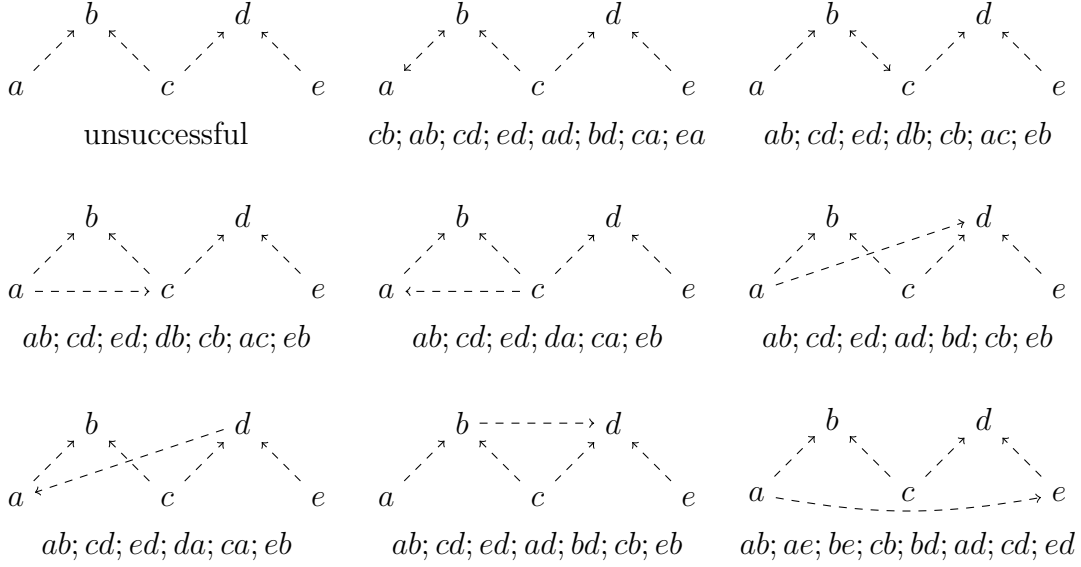


Figure 3: On the top-left a double bush of minimum size. The other figures demonstrate that adding an edge to this graph makes it weakly successful. A similar exercise demonstrates that adding a node  $f$  and an edge also makes the graph weakly successful, except when the edge is  $(f, a)$  or  $(f, e)$ , because it then still is a double bush.

$x$  be the first agent calling  $b$ , and let  $y$  be the first agent calling  $d$ . Clearly,  $x$  has to be in  $A_b$ , where we distinguish the case  $x \neq c$  from  $x = c$ .

(Case  $x \in A_b \setminus \{c\}$ ) After the call  $xb$ , agent  $x$  will not make any further call, as we can apply Lemma 36.1 to  $G_b$ . Agent  $x$  will not learn the secret of any agent in  $A_d$ , because if  $u$  calls  $x$  after  $xb$ , then  $u$  does not know the secret of  $b$ , so  $u$  must be in  $G_b$ , so that  $N^*ux$ . Thus, agent  $u$  cannot inform  $x$  of the secret of any agent not in  $A_b$ .

(Case  $x = c$ ) Consider  $G - cb + bc$ . This is a bush. Let  $\sigma'$  be the sequence obtained by replacing in  $\sigma$  call  $cb$  with  $bc$ . Then  $(G - cb + bc)^{\sigma'} = G^{\sigma}$  (and  $\sigma'$  is also maximal). By Theorem 37, LNS is unsuccessful on  $G - cb + bc$ , and so is also unsuccessful on  $G$ .  $\square$

### 5.3 Where LNS is weakly successful

In this section we prove that a gossip graph that is neither a bush nor a double bush is weakly successful for LNS. The setup of this lengthy proof is as follows.

In gossip graphs that are trees, (almost) every LNS-permitted call that is made, makes a new call LNS-permitted. We use this to define a particular LNS-sequence generating procedure, called *bottom-up call sequence* (Definition 39), that will then be used profusely in the subsequent technical results of this section. We then show that bushes and double bushes are maximal for the property of being LNS-unsuccessful: adding a new edge or a new node destroys this property (Lemmas 42–45). With these intermediary results we can then finally prove that a gossip graph that is neither a bush nor a double bush is weakly

successful for LNS (Proposition 46). The proof is by induction on the number of nodes and edges of the graph. This proof consists of many cases, of which a crucial case is supported by an additional Lemma 47.

**Definition 39 (Bottom-up call sequence)** Let  $G = (A, N, S)$  be a (possibly non-initial) gossip graph such that  $N \setminus S$  is a tree, with: root  $r$  and set of leaves  $B$ . We define a bottom-up call sequence  $\sigma(k)$  and the frontline  $B(k)$  by simultaneous induction on  $k$ . We should see the set  $B'(k)$  below as the set of successor nodes of  $B(k)$ .

$$\begin{aligned}\sigma(0) &= \epsilon \\ \sigma(k+1) &= \sigma(k); \tau(k) \\ B(0) &= B \\ B(k+1) &= B(k) \cup B'(k)\end{aligned}$$

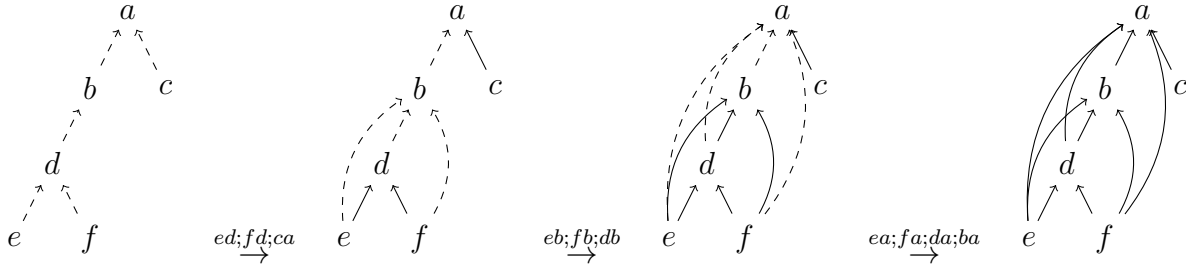
where

$$\begin{aligned}B'(k) &= \bigcup_{b \in B(k)} N_b^{\sigma(k)} \setminus S_b^{\sigma(k)} \\ \tau(k) &= \text{a maximal LNS-sequence between members of } B(k) \text{ and members of } B'(k)\end{aligned}$$

If  $n$  is the depth of the tree, then  $\sigma(k) = \sigma(n)$  for  $k > n$ . It will also be obvious that:

**Proposition 40** A bottom-up call sequence is LNS-permitted. If  $n$  is the depth of the tree, then  $\sigma(n)$  is LNS-maximal.

The graph below illustrates the execution of the bottom-up call sequence  $\tau(1); \tau(2); \tau(3)$ .



**Lemma 41** Let an initial gossip graph  $G$  be a tree with root  $r$ .

1. There is a LNS-sequence  $\sigma$  after which the root  $r$  is an expert and every agent knows the secret of  $r$ .
2. If  $r$  has exactly one predecessor, then there is a LNS-sequence that is successful on  $G$ .

**Proof** Let  $n$  be the maximum depth of  $G$ .

In order to prove the first item, we observe that in a maximal bottom-up call sequence all (other) agents have called the root  $r$ .

We now prove the second item. Let  $r'$  be the predecessor of  $r$ , and let  $G' = G - r - r'r$ . Then  $G'$  is a tree with root  $r'$ . Let  $\sigma$  be a maximal bottom-up call sequence in  $G'$ , and let

$\sigma'$  be the sequence consisting of calls from all agents except  $r$  and  $r'$ , to  $r$ . Then  $\sigma; r'r; \sigma'$  is successful on  $G$ . Now applying item 1 of this lemma to  $G'$ , we get that in  $(G')^\sigma$  agent  $r'$  knows the secrets of all agents in  $G'$ , and that all other agents in  $G'$  know the secret of  $r'$  because they have called  $r'$ . In such a call they have also learnt from  $r'$  the number of  $r$ . So, the calls in  $\sigma'$  can be made; and because they are after the call  $rr'$ , after these calls all agents in  $G$  are experts.  $\square$

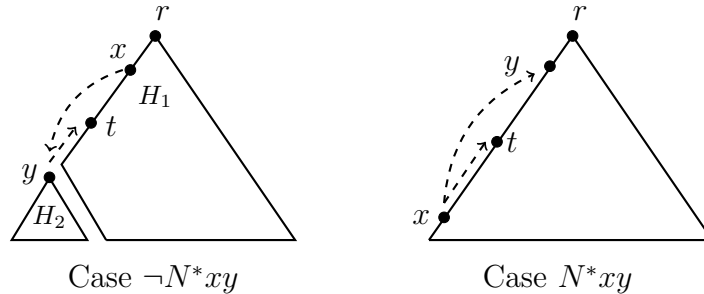
**Lemma 42** *Let  $G = (A, N, S)$  be a bush and let  $x, y \in A$  such that  $\neg Nxy$ . Then there is an LNS-sequence that is successful on  $G + xy$ .*

**Proof** If  $\neg N^*xy$ , then  $y$  is not the root  $r$  of  $G$ . Let  $t$  be the (unique)  $N$ -successor of  $y$ . Graph  $G - yt$  consists of two disjoint gossip graphs  $H_1$  and  $H_2$  with domains  $A_1$  and  $A_2$ , respectively, such that  $x, r, t \in A_1$  and  $y \in A_2$ . The generated tree with root  $x$  is a sub-graph of  $H_1$ ; let  $\sigma^x$  be a maximal bottom-up call sequence in that tree. Let  $\sigma^{xr}$  be the sequence where  $x$  calls all the agents in the path from  $x$  to  $r$  in  $H_1$ . Observe that in  $H_1^{\sigma^x; \sigma^{xr}}, N_1^{\sigma^x; \sigma^{xr}} \setminus S_1^{\sigma^x; \sigma^{xr}}$  is a tree (Lemma 36.1). Let  $\sigma^1$  and  $\sigma^2$  be maximal bottom-up call sequences for  $H_1^{\sigma^x; \sigma^{xr}}$  and  $H_2$ , respectively. The call sequence

$$\sigma^x; \sigma^{xr}; \sigma^1; \sigma^2; ry; (A_1 - r)y; (A_2 - y)t$$

is LNS-permitted and successful on  $G + xy$ ; where, as usual,  $(A_i - u)w$  is a sequence of calls from each agent in  $A_i$  except  $u$ , to agent  $w$ . After  $\sigma^x; \sigma^{xr}; \sigma^1$ , the root  $r$  is an expert in  $H_1$  and all the agents in  $H_1$  have the number of  $y$ . After  $\sigma^2; ry$ , both  $r$  and  $y$  are experts and all agents in  $H_2$  have the number of  $t$ . After  $(A_1 - r)y$  all agents in  $A_1$  are experts, in particular  $t$ . At the end, all agents are experts.

If  $N^*xy$ , then, since  $x \neq y$ ,  $x$  is not the root. Let  $t$  be the  $N$ -successor of  $x$ . Write  $G + xy$  as  $(G - xt + xy) + xt$ . Graph  $(G - xt + xy)$  is a bush, and there is no path from  $x$  to  $t$  in that graph. We therefore can apply the part ‘if  $\neg N^*xy$ ’ of this proof, on  $(G - xt + xy)$  and  $xt$ , instead of on  $G$  and  $xy$ .



$\square$

**Lemma 43** *Let  $G = (A, N, S)$  be a bush,  $y \notin A$  a new node,  $x \in A$ , and let  $e$  be an edge between  $y$  and  $x$ . If  $G + y + e$  is not a bush, there is an LNS-sequence that is successful on  $G$ .*

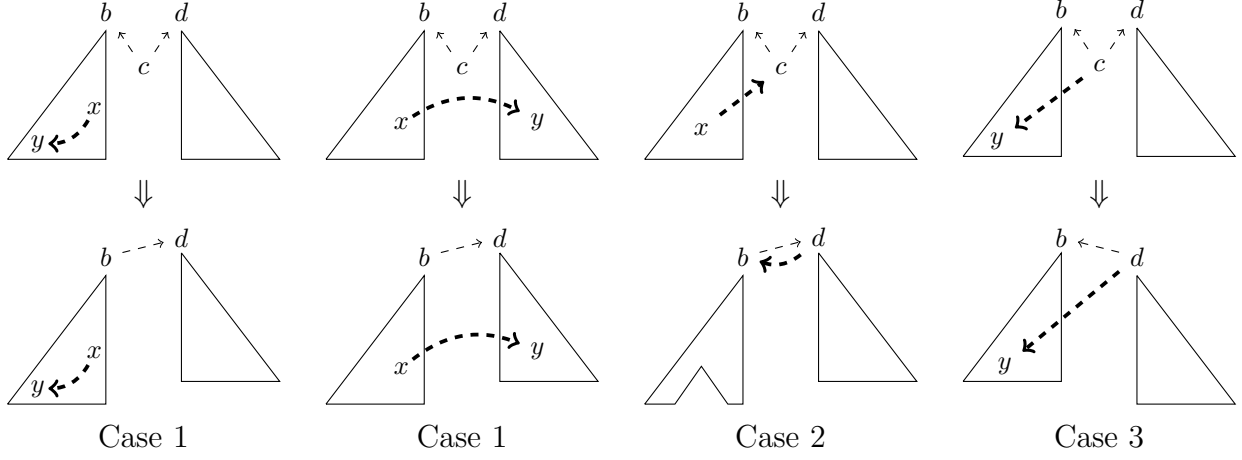


Figure 4: Three cases used in the proof of Lemma 44 (top line), and their modifications (bottom line). If  $y \neq c$  (Case 1),  $y$  can be in part  $G_b$  (left) but also in part  $G_d$  (right) of the graph.

**Proof** If  $e = yx$  then  $G + y + yx$  is again a bush, in which case the lemma is trivially true. So let  $e = xy$ . If  $x = r$  we can use Lemma 41.2, so we can additionally assume that  $x \neq r$ . Let  $H$  be  $G + y + yx$ . Then,  $H$  is a bush. Hence, we can apply Lemma 42 to obtain a successful LNS-sequence on  $H + xy$ , say  $\sigma$ . If  $\sigma$  does not contain the call  $yx$ , then  $\sigma$  is also LNS-permitted in  $G + y + xy$ , and hence also successful. Otherwise, in case  $\sigma$  contains the call  $yx$ , simply replace it with the call  $xy$ . Clearly, the resulting sequence  $\sigma'$  is again LNS-permitted in  $G + y + xy$  and also successful.  $\square$

**Lemma 44** *Let  $G$  be a double bush, and let  $\neg Nxy$ . Then  $G + xy$  has a successful LNS execution.*

**Proof** Let  $G = (A, N, S)$ . Recall the different components of a double bush with roots  $b, d$  and a common node  $c$ :  $A = A_b \cup A_d$  and  $A_b \cap A_d = \{c\}$ . As these components are alike, it suffices to consider 3 cases (see Figure 4).

(Case 1:  $x \in A^b - c$  and  $y \neq c$ .) In this case  $y$  may be in part  $G_b$  or in part  $G_d$  of the graph, but that does not matter for the proof.

If  $xy = bd$ , then  $G - cb + bd$  is a bush, so by Lemma 42, we obtain a successful LNS-sequence  $\sigma$  on  $(G - cb + bd) + cb$ , i.e., on  $G + bd$ .

If  $xy \neq bd$ , consider  $H = G - c - cb - cd + bd$ . As  $H$  is a bush, we can apply Lemma 43 to obtain an LNS-sequence  $\sigma$  successful on  $H + xy$ . The sequence  $cb; \sigma; cd$  is LNS-permitted and successful on  $G + xy = (H + xy) + c + cb + cd - bd$ .

(Case 2:  $x \in A^b - c$  and  $y = c$ .) If  $x = b$ ,  $G - cb + bc$  is a bush, so we can apply Lemma 42 to obtain a successful LNS-sequence  $\sigma$  on  $(G - cb + bc) + cb = G + bc$ . If  $x \neq b$ , let  $G^x$ , for agents  $A^x$ , be the subtree of  $G$  with root  $x$ , and let  $\sigma^x$  be a maximal bottom-up call sequence on  $G^x$ . The graph  $H = G - G^x - c - cb - cd + bd$  is a bush, so we can apply Lemma 42 to obtain a successful LNS-sequence  $\sigma$  on  $H + db$ . On  $G + xc$ , sequence

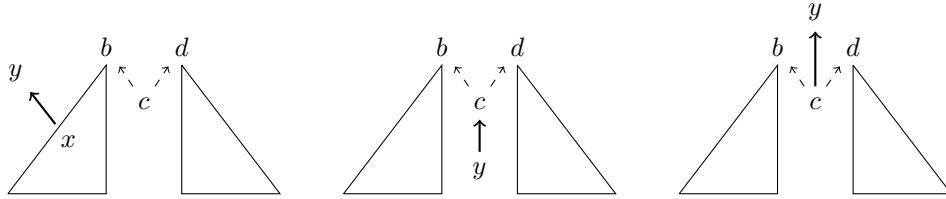


$\sigma^x; xc; cb; xd; \sigma; cd; xb; (A^x - x)c$  is LNS-successful, where, as before,  $(A^x - x)c$  denotes a sequence from all agents in  $(A^x - x)c$  to agent  $c$ . From Lemma 41 it follows that after  $\sigma^x$  everybody knows the secret of  $x$  and thus also the number of  $c$ , so the calls  $(A^x - x)c$  can indeed take place.

(Case 3:  $x = c$  and  $y \in A^b$ .) Let  $H = G - c - cb - cd + db$ . This is a bush. We apply Lemma 42 to obtain a successful LNS-sequence  $\sigma$  on  $H + dy$ . On  $G + cy$  the sequence  $cd; \sigma; cb$  is LNS-successful.  $\square$

**Lemma 45** *Let  $G = (A, N, S)$  be a double bush,  $y \notin A$ ,  $x \in A$ , and  $e$  an edge between  $x$  and  $y$ . There is a successful LNS call sequence on  $G + y + e$  unless it is a double bush.*

**Proof** Let  $A = A_b \cup A_d$  and  $A_b \cap A_d = c$ , where  $b, d$  are the two roots and  $c$  is the common leaf. Without loss of generality, suppose  $x \notin A_d$ . Consider the bush  $H = G - c - cb - cd + bd$ . There are 3 cases to consider in this proof:  $e = xy$ ,  $e = yc$ , and  $e = cy$ .



If  $x \in A^b - c$  and  $e = xy$ , apply Lemma 43 on  $H$  to obtain a successful LNS-sequence  $\sigma$  on  $H + y + xy$ . The sequence  $cb; \sigma; cd$  is LNS-permitted and successful on  $G + y + xy$ .

If  $x = c$  and  $e = yc$ , apply Lemma 42 to obtain a successful LNS-sequence  $\sigma$  on  $H + db$ . The sequence  $yc; cb; yd; \sigma; cd; yb$  is LNS-permitted and successful on  $G + y + yc$ .

If  $x = c$  and  $e = cy$ , apply Lemma 43 to obtain a successful LNS-sequence  $\sigma$  on  $H + y + by$ . The sequence  $cb; \sigma; cd$  is LNS-permitted and successful on  $G + y + cy$ .  $\square$

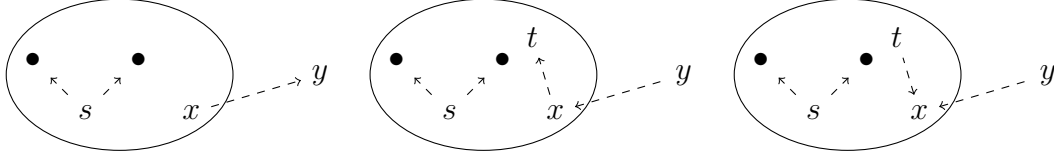
**Proposition 46** *Let  $G$  be a weakly connected graph. If  $G$  is neither a bush nor a double bush, then  $G$  is weakly successful on LNS.*

**Proof** The proof is by strong induction on the sum of the number  $n$  of nodes and the number  $m$  of edges (not counting loops) of such gossip graphs  $G$ . If  $n + m = 1$ ,  $G$  is a single agent gossip graph, on which the empty sequence is successful. Let now  $G$  be a gossip graph with  $n + m = k + 1$  nodes and edges, and assume the proposition to be proved for  $l \leq k$  nodes and edges (call such an inductive case  $\varphi(l)$ ).

If  $G$  has an edge  $e$  which is not a bridge, then  $G - e$  is weakly connected. In case this  $G - e$  is neither a bush nor a double bush, then by inductive hypothesis (for  $k$ ) there is a successful LNS-sequence  $\sigma$  on  $G - e$ . This  $\sigma$  is also successful on  $G$ . In case  $G - e$  is a bush we apply Lemma 42, and in case  $G - e$  is a double bush we apply Lemma 44, in order to obtain a successful LNS-sequence  $\sigma$  on  $(G - e) + e = G$ .

Let now every edge in  $G$  be a bridge. If  $G$  does not have a node with out-degree greater than 1, then  $G$  is a tree. The input-degree of the root of this tree must be 1, because we assumed that  $G$  is not a bush. By Lemma 41.2 there must be a successful LNS sequence

on  $G$ . However, if  $G$  has a node  $s$  with output degree greater than 1, then, since the undirected graph underlying  $G$  is a tree, there are at least two nodes with degree 1. Let  $y$  be such a node with degree 1. We need to consider three cases (pictured from left to right, itemized from top to bottom):



- If there is an edge  $xy$  with  $\deg_{\text{out}}(y) = 0$  and  $\deg_{\text{in}}(y) = 1$ , consider  $G - y - xy$ . If  $G - y - xy$  is neither a bush nor a double bush, then by inductive hypothesis for  $k - 1$  there is a successful LNS-sequence  $\sigma$  on  $G - y - xy$ . Let now  $\tau$  be a sequence where all agents in  $G$  except  $y$  call  $y$ . We then have that  $\sigma; \tau$  is a successful LNS sequence on  $G$ . If however  $G - y - xy$  is a bush, then by Lemma 43 we obtain a successful LNS sequence on  $G$ ; and if  $G - x - xy$  is a double bush, we obtain this by Lemma 45.
- Let there be an edge  $yx$  with  $\deg_{\text{in}}(y) = 0$  and  $\deg_{\text{out}}(y) = 1$  and  $\deg_{\text{out}}(x) \geq 1$ . If  $G - y - yx$  is neither a bush nor a double bush, then by inductive hypothesis for  $k - 1$  there is a successful LNS-sequence  $\sigma$  on  $G - y - yx$ , which can be extended into a sequence  $yx; \sigma; yt$  that is LNS-successful on  $G$ , where  $t$  is the successor of  $x$  (see the figure above). Otherwise we proceed as before by applying Lemma 43 or Lemma 45.
- Lemma 47 handles the remaining and more complex case where, for any node  $y$ , if  $\deg_{\text{io}}(y) = 1$ , then:  $\deg_{\text{out}}(y) = 1$  and the successor  $x$  of  $y$  is a terminal; as above on the right. Provided that  $x$  is terminal, since  $G$  is not a bush, neither it will be  $G - y - yx$ ; similarly we conclude that  $G - y - yx$  is not a double bush. So we can apply the inductive hypothesis to obtain some sequence  $\sigma$  successful on  $G - y - yx$ . But the problem is to modify  $\sigma$  into a solution for  $G$ .

□

We introduce some more terminology, to be used in Lemma 47. A *source* is a node with  $\deg_{\text{out}}(s) \geq 2$ . An *initial source* is a source that is a minimal node. A *t-ghost* (or *ghost*) is a node  $x$  with  $\deg_{\text{in}}(x) = 0$ ,  $\deg_{\text{out}}(x) = 1$ , and  $Nxt$  for some terminal node  $t$ . We introduce some notation for a path  $x_1, \dots, x_n$  as  $[x_1, x_n]$ ; and where  $x \in [x_1, x_n]$  means  $x \in \{x_1, \dots, x_n\}$ . If we want to exclude the first node in the path we write  $(x_1, x_n]$ , and similarly for  $[x_1, x_n)$  and (unless this causes ambiguity)  $(x_1, x_n)$ . A *lonely path* is a path  $[x_1, x_n]$  such that for all  $y \in (x_1, x_n)$ , the in-degree and out-degree of  $y$  is 1. In other words, in a lonely path there is no branching in or branching out, except maybe at the first or at the last node.

Let  $G = (A, N, S)$  be a gossip graph,  $B \subseteq A$  and  $x \notin A$ . The  $B$ -*abstraction* of  $G$  is the graph  $G' = (A', N', S')$  where:  $A' = (A \setminus B) + x$ ;  $N'y z$  iff  $(Nyz, \text{ or } x = y \text{ and } \exists w \in B \text{ such that } Nwz, \text{ or } x = z \text{ and } \exists w \in B \text{ such that } Nyw, \text{ or } x = y = z)$ ; and similarly for  $S'$ .

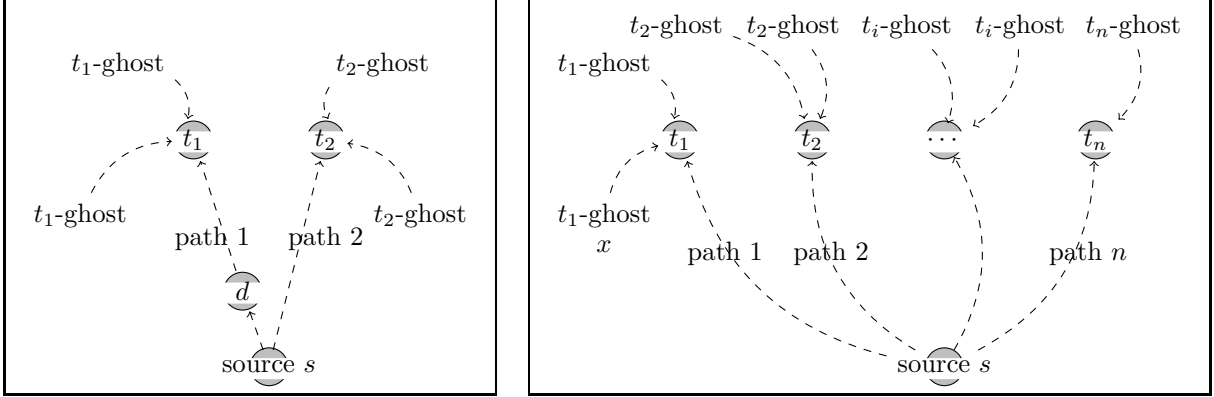


Figure 5: Gossip graphs with one source node, with 2 and with more than 2 successors

**Lemma 47** *Let  $G$  be a weakly connected gossip graph where all edges are bridges, with at least one source node, wherein any node with degree 1 is a ghost, and that is not a double bush. Then  $G$  has a successful LNS sequence.*

**Proof** The proof is by induction on the number of source nodes. There are two base cases, namely for 1 source node, and another one for 2 source nodes in a particular configuration (defined in that case). The inductive case applies to (all at the same time) 2 source nodes not in that particular configuration and 3 or more source nodes.

**(One source node)**

The treatment for a single source with two successors is different from the treatment for a single source with more than two successors. Figure 5 illustrates the two cases. Terminals are named  $t_1, t_2, \dots, t_n$  and the source is named  $s$ . Node  $d$  is the successor of  $s$ . The  $t_1$ -ghost on the right called  $x$  plays a role in the proof.

If  $\deg_{\text{out}}(s) = 2$  at least one path must have length 2, because otherwise the graph is a double bush; say this is on the path  $[s, t_1]$ , and that the successor of  $s$  is  $d$ . This following call sequence is LNS-permitted and successful on  $G$ :

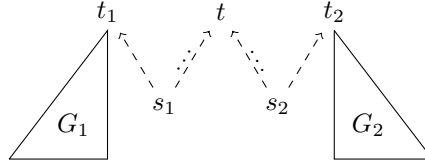
- a maximal bottom-up call sequence for  $[s, t_2]$
- call  $sd$ ;
- a maximal bottom-up call sequence for  $[d, t_1]$ ;
- each  $t_1$ -ghost calls  $t_1$ ;
- call  $st_2$ ; each  $t_2$ -ghost calls  $t_2$ ;
- call  $t_1t_2$ ; all agents in path  $(s, t_1)$  and all  $t_1$ -ghosts call  $t_2$ ;
- all agents in path  $[s, t_2]$  and all  $t_2$ -ghosts call the successor of  $d$ ;

If  $\deg_{\text{out}}(s) \geq 3$ , fix  $x$  to be some  $t_1$ -ghost. The call sequence consisting of successively the following subsequences is LNS-successful on  $G$ :

- a maximal bottom-up call sequence for path  $[s, t_2]$ ;
- a maximal bottom-up call sequence for path  $[s, t_1]$ ;
- each  $t_1$ -ghost calls  $t_1$ ;
- call  $st_2$ ; each  $t_2$ -ghost calls  $t_2$ ;
- for  $3 \leq i \leq n$ :  $t_1$  calls any agent along the path  $(s, t_i]$ ; each  $t_i$ -ghost calls  $t_i$ ;  $x$  calls any agent along the path  $(s, t_i]$ ;
- $t_n$  calls  $t_2$ ; all agents except  $s, t_n, t_2$ , and  $t_2$ -ghosts call  $t_2$ ;
- the source  $s$  and each  $t_2$ -ghost call the successor of  $s$  in path  $n$ .

**(Two source nodes with special condition)**

Let  $G = G_1 + G_2 + G_3$  be a gossip graph such that  $G_1$  and  $G_2$  are trees with root  $t_1$  and  $t_2$ , a source  $s_1$  is connected by a link to  $t_1$  and by a path to a terminal  $t$ , and a source  $s_2$  is connected by a link to  $t_2$  and by a path to a terminal  $t$ .



Let  $\sigma^1$  and  $\sigma^2$  be maximal bottom-up call sequences in, respectively,  $G_1$  and  $G_2$ . The sequence

$$[s_1 t]; [s_2 t]; tt_1; s_2 t_2; \sigma^1; \sigma^2; t_2 t_1; [s_1, t) t_1; [s_2, t) t_1; tt_2; \tau^1; \tau^2$$

is LNS-successful on  $G$ , where  $[s_1 t]$  is a maximal bottom-up sequence for path  $[s_1, t]$ , and similarly for  $[s_2 t]$ , where  $[s_1, t) t_1$  is a sequence wherein each agent in  $[s_1, t)$  calls  $t_1$  (and  $[s_2, t) t_1$  is defined similarly); finally,  $\tau^1$  is the sequence that all the agents in  $G_1$  except  $t_1$  call  $t_2$ ;  $\tau^2$  is the sequence that all the agents in  $G_2$  except  $t_2$  call  $t_1$ .

**(At least two source nodes)**

We first assume that there are two initial source nodes  $s_1$  and  $s_2$ . Let  $u$  be the node where the paths from  $s_1$  and  $s_2$  meet (so  $[s_1, u]$  and  $[s_2, u]$  are lonely paths—no branching in, no branching out).

Let  $\sigma^1$  and  $\sigma^2$  be maximal bottom-up call sequences in, respectively,  $[s_1, u]$  and  $[s_2, u]$ . If  $u$  is a terminal, then let  $\sigma^3$  be a call sequence from each  $u$ -ghost to  $u$ ; otherwise  $\sigma^3 = \epsilon$ . Consider the  $([s_1, u] \cup [s_2, u] \cup \{z \mid z \text{ is a } u\text{-ghost}\})$ -abstraction of  $G$  and let  $x$  be the new node generated by this abstraction. Call this graph  $G'$ . Graph  $G'$  has one less source. If  $G'$  is a double bush, the base case **(Two source nodes with special condition)** applies. Otherwise, by inductive hypothesis there is a successful LNS sequence  $\sigma$  on  $G'$ .

Let now  $v$  be a successor of  $s_1$  that is not in the path  $[s_1, u]$ . Let  $\tau$  be obtained from  $\sigma$  by replacing each occurrence of  $x$  with  $u$ , and let  $\tau'$  be a call sequence where all agents in

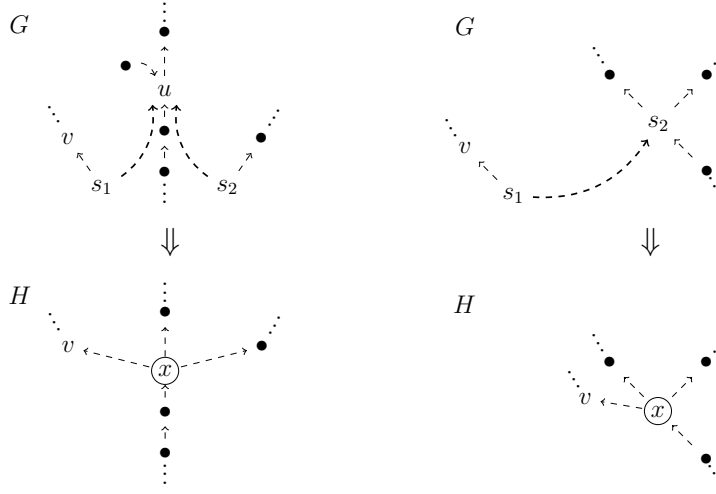


Figure 6: On the left: two initial sources. On the right: an initial and a non-initial source

$[s_1, u) \cup [s_2, u)$ , and also all  $u$ -ghosts, call  $v$  (that is, all the agents different from  $u$  removed in the abstraction call  $v$ .)

Then a successful LNS call sequence on  $G$  is  $\sigma^1; \sigma^2; \sigma^3; \tau; \tau'$ .

We can further justify why it is successful. After  $\sigma^1; \sigma^2; \sigma^3$ , agent  $u$  knows the secret of all agents in  $[s_1, u) \cup [s_2, u) \cup \{z \mid z \text{ is a } u\text{-ghost}\}$ . After  $\tau$ , all agents in  $G$  are experts except those in the abstracted part that are not  $u$ . After  $\tau'$ , those other agents also become experts.

If there are no two initial sources, then, given that there are at least two sources, there must be two sources  $s_1, s_2$  such that  $s_1$  is an initial source and  $s_1$  is connected to  $s_2$  by a lonely path. Let  $\sigma^1$  be a maximal bottom-up call sequence in  $[s_1, s_2]$ . Consider the  $[s_1, s_2]$ -abstraction of  $G$  and let  $x$  be the new node. Let this graph be  $H$ . Since  $H$  has one less source node (and since it is not a double bush), there is an LNS-sequence  $\sigma$  successful on  $H$ . Let  $\tau$  be the call sequence obtained from  $\sigma$  where we replace each occurrence of  $x$  with  $s_2$ , and let  $\tau'$  be the call sequence where every agent in  $[s_1, s_2)$  calls a successor  $v$  of  $s_1$  not in  $[s_1, s_2]$ . Similarly to the previous case, it can be easily shown that  $\sigma^1; \tau; \tau'$  is an LNS-sequence successful on  $G$ .  $\square$

This finally brings us to our main result:

**Theorem 48** *LNS is weakly successful on a weakly connected gossip graph  $G$  iff  $G$  is neither a bush nor a double bush.*

**Proof** From Proposition 37 (page 20), Proposition 38 (page 20), and Proposition 46 (page 25).  $\square$

## 6 Conclusions and Further Research

**Conclusions** We investigated distributed gossip protocols where not only secrets are exchanged but also telephone numbers, such that the exchange of numbers leads to expansion of the so-called gossip graph, our representation of the communication network. We defined various such distributed dynamic gossip protocols, and we characterized them in terms of the class of graphs where they terminate successfully, and this we did for three different termination conditions: strong success (all protocol executions terminate with all agents knowing all secrets), fair success (all fair executions terminate), and weak success (at least one execution terminates). In the Any Call protocol (**ANY**) agents can make any call. In the Token protocol (**TOK**) the agent who is called is allowed to make a next call, whereas in the Spider protocol (**SPI**) the agent who is calling is allowed to make a next call. In the Call Me Once protocol (**CO**) agents cannot call someone again if they already were involved in a prior call, whereas in the Weak Call Me Once protocol (**wCO**) the restriction only applies if you were the caller in that call, not if you were the callee. The Learn New Secrets protocol (**LNS**) only allows agents to call other agents whose secret they do not know. These are the characterization results (we recall that **ANY**, **TOK**, and **SPI** are in fact strongly successful on weakly connected two-agent graphs, but the answer to the gossip problem for the class of *all* finite initial gossip graphs is ‘no’):

	ANY, TOK, SPI	CO, wCO	LNS
<i>strong success</i>	no	weakly connected graph ( <i>Theorem 25, page 14</i> )	sun graph ( <i>Theorem 34, page 17</i> )
<i>fair success</i>	weakly connected graph ( <i>Theorems 17, 22, 23</i> )	„	„
<i>weak success</i>	„	„	no (bush or double bush) ( <i>Theorem 48, page 29</i> )

**Protocol extension hierarchy** It may be of interest to the reader to compare the extensions of the protocols presented in this paper. These protocol extensions determine a partial order on the class of all gossip graphs. We have that  $\text{LNS} \subset \text{CO} \subset \text{wCO} \subset \text{ANY}$ , and that **SPI** and **TOK** are incomparable and both of them are also incomparable to any of **LNS**, **CO**, and **wCO**. These results are depicted in Figure 7.

Let us explain Figure 7 in some detail. Each area in the figure represents the set of call sequences for all gossip graphs, that are determined in the obvious way by algebraic manipulation of protocol extensions. As representatives of these sets we have chosen call sequences for the initial gossip graph consisting of three agents  $a, b, c$  such that  $Nab$  and  $Nbc$ . For example, the area containing  $ab;ba$  defines  $\text{TOK} \cap \text{wCO} \cap \overline{\text{CO}} \cap \overline{\text{SPI}}$ . This means that  $ab;ba$  is also an **ANY** call sequence, but not a **CO** (and therefore also not a **LNS**) or **SPI** call sequence (call  $ba$  is then not permitted after  $ab$ ). The sequence  $ab;ab;ba$  is an **ANY** call sequence that is not permitted in any other of the protocols defined in the paper. If an area contains ‘—’, then the corresponding extension is empty, and not only for the example gossip graph, but for all gossip graphs. For example,  $\overline{\text{LNS}} \cap \text{SPI} \cap \text{TOK} = \emptyset$ . Indeed, a

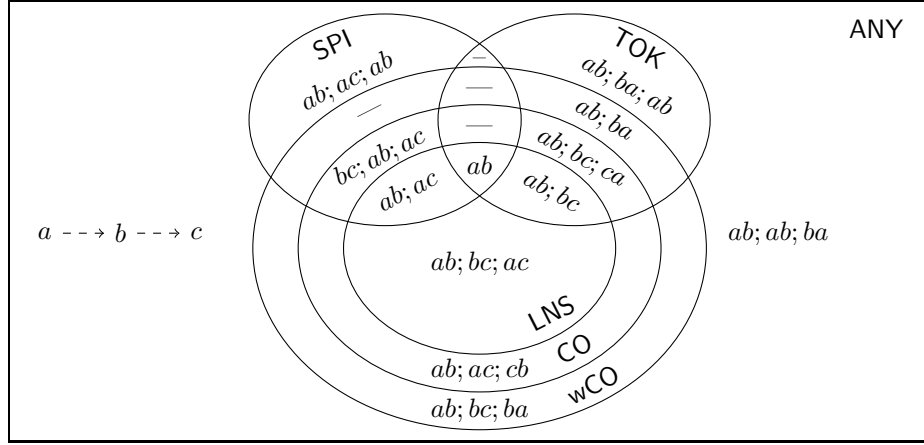


Figure 7: The protocol extension hierarchy, illustrated for the gossip graph on the left

call sequence  $\sigma$  that is not in **LNS** cannot be both in **SPI** and **TOK**: let  $xz$  be the first not **LNS**-permitted call in  $\sigma$ , there must then be a previous call involving  $x$  in  $\sigma$ ; if that call was from  $x$ , then  $xz$  is not **TOK**-permitted, but if the call was to  $x$ , then  $xz$  is not **SPI**-permitted.

**Further research** Of course many other dynamic distributed protocols than those presented in this paper are conceivable. We have investigated some other protocols, but their characterizations were either the same or were not sufficiently interesting to present in-depth. For computational or optimizational reasons such other protocols would come back into the picture. Computational aspects of dynamic gossip protocols are investigated in [12], in progress. Of great interest to us seem combinatorial results on expected protocol execution length, for which there are to our knowledge few results (some simulation results for **LNS** are mentioned in [2]). A comparative analysis of expected execution length for the protocols presented here and yet other protocols seems of great interest.

Apart from exchanging numbers and secrets, one can also consider one-way traffic, where only the caller or only the callee provides information, or where only some but not all secrets are exchanged (as in [1], but then for the dynamic gossip setting). Explicit representation of knowledge, both for synchronous and asynchronous settings (as in [3, 1]) seems also of theoretical interest, in order to build bridges to the logical or distributed systems communities.

## References

- [1] K. R. Apt, D. Grossi, and W. van der Hoek. Epistemic protocols for distributed gossiping. In *Proceedings of 15th TARK*, 2015.

- [2] M. Attamah. *Epistemic gossip protocols*. PhD thesis, Computer Science, The University of Liverpool, 2015.
- [3] M. Attamah, H. van Ditmarsch, D. Grossi, and W. van der Hoek. Knowledge and gossip. In *Proc. of 21st ECAI*, pages 21–26. IOS Press, 2014.
- [4] P. T. Eugster, R. Guerraoui, A. Kermarrec, and L. Massoulié. Epidemic information dissemination in distributed systems. *IEEE Computer*, 37(5):60–67, 2004.
- [5] R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning about Knowledge*. MIT Press, Cambridge MA, 1995.
- [6] M. C. Golumbic. The general gossip problem. *Technical Report IBM Research Report RC4977, IBM*, 1974.
- [7] A. Hajnal, E. C. Milner, and E. Szemerédi. A cure for the telephone problem. *Canadian Mathematical Bulletin*, 15(3):447–450, 1972.
- [8] F. Harary and A. J. Schwenk. The communication problem on graphs and digraphs. *Journal of Franklin Institute*, 297:491–495, 1974.
- [9] S. M. Hedetniemi, S. T. Hedetniemi, and A. L. Liestman. A survey of gossiping and broadcasting in communication networks. *Networks*, 18(4):319–349, 1988.
- [10] C. A. J. Hurkens. Spreading gossip efficiently. *NAW*, 5(1):208–210, 2000.
- [11] R. Tijdeman. On a telephone problem. *Nieuw Archief voor Wiskunde*, 3(19):188–192, 1971.
- [12] J. van Eijck and M. Gattling. Gossip. Technical Report, CWI, Amsterdam, available from [www.cwi.nl/~jve/papers/15/pdfs/Gossip.pdf](http://www.cwi.nl/~jve/papers/15/pdfs/Gossip.pdf), 2015.